

RANCANG BANGUN PENGAMATAN PH AIR MENGUNAKAN PROTOKOL MQTT

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Cahyo Karya Basuki
135150207111108



TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2017

PENGESAHAN

RANCANG BANGUN PENGAMATAN PH AIR MENGGUNAKAN PROTOKOL MQTT

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Cahyo Karya Basuki
135150207111108

Skripsi ini telah diuji dan dinyatakan lulus pada
2 Agustus 2018

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I



Dany Primanita Kartikasari, ST, M.Kom
NIP. 19771116 200501 2 003

Dosen Pembimbing II



Achmad Basuki, ST, M.MG, Ph.D
NIP. 19741118 200312 1 002

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T, M.T, Ph.D

NIP. 19710518 200312 1 001

IDENTITAS PENGUJI

1. Dosen Penguji I

Kasyful Amron, S.T, MSc

NIP. 19750803 200312 1 003

2. Dosen Penguji II

Ir. Heru Nurwarsito, M.Kom

NIP. 19650402 199002 1 001



PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, dinaskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata dinaskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 2 Agustus 2018



Cahyo Karya Basuki

NIM: 135150207111108

DAFTAR RIWAYAT HIDUP

Nama : Cahyo Karya Basuki

Tempat, Tanggal Lahir : Situbondo, 06 Februari 1995

Alamat Asal : Jl. Jatibanteng Ds. Bloro-Pettak RT 12 RW 05
Besuki, Situbondo

Nama Orang Tua : Waras

Riwayat Pendidikan : SDN 1 Jatibanteng (2001-2007)
SMP Negeri 1 Banyuglugur (2007-2010)
SMA Negeri 1 Panji (2010-2013)
S1 Informatika Universitas Brawijaya (2013-2018)

Alamat di Malang : Mt. Haryono Gang 13 no 438 Dinoyo

No. telpon/HP : 082315023501

E-mail : cahyokarya06@gmail.com

Prestasi : -

Pengalaman Kepanitiaan : -

Pengalaman Organisasi : -

UCAPAN TERIMA KASIH

Ucapan terima kasih penulis haturkan kepada seluruh pihak yang selama ini telah mendukung dan membantu dalam proses penelitian skripsi. Kepada dosen pembimbing I dan dosen pembimbing II yang telah meluangkan waktunya dalam membimbing dan memberikan masukan berharga bagi skripsi ini. Ucapan terima kasih juga penulis haturkan kepada teman-teman di kampus yang telah memberikan dukungan dan berjuang bersama-sama dalam menyelesaikan penelitian. Semoga segala jerih payah perjuangan yang telah dilakukan bisa memberi manfaat bagi banyak orang.

Malang, 2 Agustus 2018

Penulis

Cahyokarya06@gmail.com



KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, karena berkat rahmat serta bimbingan-Nya, penulis dapat menyelesaikan skripsi yang berjudul “RANCANG BANGUN PENGAMATAN PH AIR MENGGUNAKAN PROTOKOL MQTT” dengan baik. Sebagai persyaratan untuk memperoleh gelar Sarjana Komputer pada Proram Studi Teknik Informatika Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.

Dalam penyusunan dan penulisan skripsi ini tidak terlepas dari bantuan, bimbingan serta dukungan dari berbagai pihak. Oleh karena itu dalam kesempatan ini penulis dengan senang hati menyampaikan terima kasih kepada:

1. Bapak Waras dan Ibu Sukarna selaku kedua orangtua penulis yang selalu memberikan dukungan dan doa yang tidak pernah lelah.
2. Ibu Dany Primanita Kartikasari, ST, M.Kom selaku dosen pembimbing I yang telah dengan sabar, tulus, ikhlas dan tekun dalam meluangkan waktu, tenaga dan pikiran dalam memberikan bimbingan, motivasi, arahan dan saran terhadap penulis selama menyusun skripsi.
3. Bapak Achmad Basuki, ST, M.MG, Ph.D selaku dosen pembimbing II yang telah dengan sabar, tulus, ikhlas dan tekun dalam meluangkan waktu, tenaga dan pikiran dalam memberikan bimbingan, motivasi, arahan dan saran terhadap penulis selama menyusun skripsi.
4. Rani Dewi Asmarani yang selalu memberikan doa, dukungan serta motivasi untuk penulis.
5. Teman–teman angkatan 2013 Informatika, geng beng, terima kasih atas segala bantuan selama menempuh studi di Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
6. Serta seluruh pihak yang dengan ikhlas membantu dan meluangkan waktunya untuk penulis agar menyelesaikan skripsi yang tidak bisa ditulis satu persatu.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan. Penulis mengharapkan adanya saran dan kritik membangun dari para pembaca demi kesempurnaan skripsi ini. Semoga skripsi ini dapat memberikan manfaat bagi semua dan berguna bagi pengembangan ilmu pengetahuan.

Malang, 2 Agustus 2018

Penulis

Cahyokarya06@gmail.com

ABSTRAK

pH air merupakan aspek terpenting dalam bidang perikanan, terutama pada pembudidayaan ikan dalam kolam. Masalah pemantauan pH, saat ini sangat banyak dibicarakan salah satunya adalah perubahan cuaca. Kondisi cuaca yang tidak diharapkan oleh pihak pembudidaya terkadang menjadi sebuah permasalahan. Permasalahan tersebut berdampak pada waktu pemrosesan dalam pengambilan sampel pH air. Sehingga dibutuhkan peranan teknologi yang dapat membantu mengatasi masalah tersebut, untuk menyampaikan data pH air kepada pengelola dari jarak jauh. IoT atau *Internet of Things* merupakan sebuah konsep teknologi yang penerapannya dapat digunakan untuk membantu mengatasi masalah tersebut. Sistem kerja dari IoT adalah menghubungkan benda – benda fisik dengan komunikasi internet. Untuk saling terhubung satu sama lain, perangkat – perangkat yang digunakan memerlukan sebuah bantuan protokol. Protokol tersebut nantinya menjalankan transmisi data yang berfungsi sebagai transportasi data dari sensor kepada pengguna dalam sebuah jaringan. Protokol yang digunakan adalah MQTT (*Message Queuing Telemetry Transport*) dimana protokol tersebut diharapkan dapat bekerja dalam kondisi *low bandwidth* serta *unreliable network*. Protokol MQTT yang digunakan mengadopsi mekanisme *publish-subscribe*.

Dengan demikian sistem yang dibuat terdiri dari 3 komponen yaitu *publisher*, *broker* dan *subscriber*. Komponen pertama adalah *sensor node* pada mikrokontroler arduino sebagai *publisher*. Kemudian komponen kedua adalah mikrokomputer Raspberry Pi sebagai *server* yang berfungsi sebagai *broker*. Didalam *server* terdapat *middleware* pengontrol waktu dan penyimpanan data. Komponen ketiga adalah *subscriber* yang mengadopsi dari bahasa *java* sebagai *interface* kepada pengguna. Dari implementasi yang dibuat memperoleh hasil pengujian sistem yang sesuai harapan dan memperoleh kinerja protokol yang merujuk kepada nilai *average latency* dan *delay* dibawah 1 ms yaitu 0,31 ms dan 0,71 ms dengan lama waktu uji 3x24 jam.

Kata kunci: pH air, MQTT, Publish Subscribe, Raspberry Pi, Middleware.

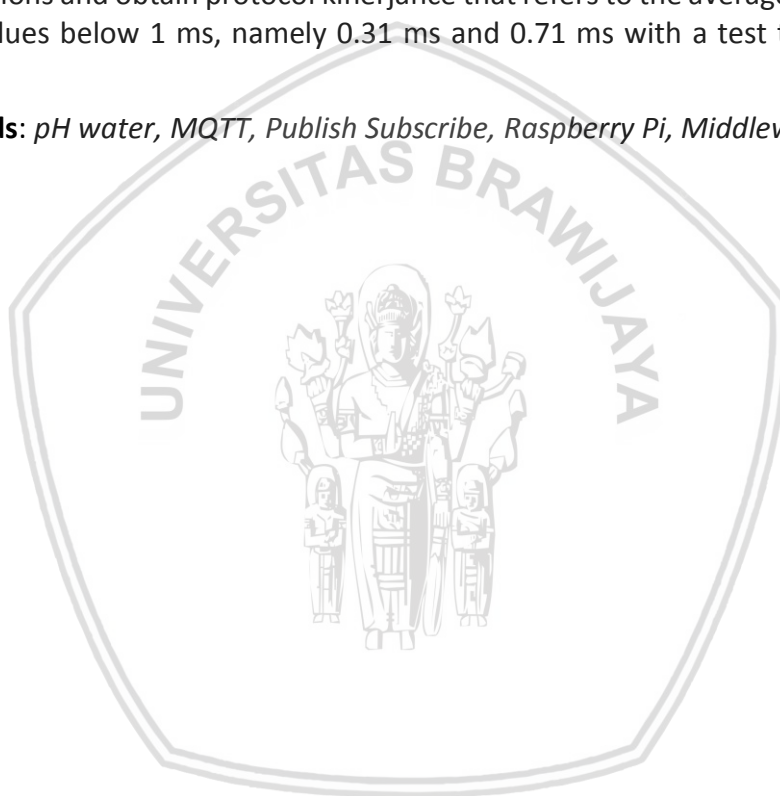
Abstract

Water pH is the most important aspect in the field of fisheries, especially in fish cultivation in ponds. PH monitoring problems, currently very much discussed one of them is weather changes. Weather conditions that are not expected by the farmers sometimes become a problem. These problems have an impact on the processing time in taking pH water. So that the role of technology is needed that can help overcome the problem, to convey water pH data to managers remotely. IoT or the Internet of Things is a technology concept that its application can be used to help overcome these problems. The work system of IoT is connecting physical objects with internet communication. To connect to each other, the devices used need a protocol help. The protocol will then run data

transmission that functions as data transport from sensors to users on a network. The protocol used is MQTT (Message Queuing Telemetry Transport) where the protocol is expected to work in low bandwidth and unreliable networks. The MQTT protocol used adopts the publish-subscribe mechanism.

Thus the system created consists of 3 components, namely publishers, brokers and subscribers. The first component is the sensor node on the Arduino microcontroller as a publisher. Then the second component is the Raspberry Pi microcomputer as a server that functions as a broker. In the server there is a time control and data storage middleware. The third component is subscribers who adopt from the Java language as an interface to the user. From the implementation made to obtain the results of system testing that match expectations and obtain protocol kinerjance that refers to the average latency and delay values below 1 ms, namely 0.31 ms and 0.71 ms with a test time of 3x24 hours.

Keywords: *pH water, MQTT, Publish Subscribe, Raspberry Pi, Middleware.*



DAFTAR ISI

PENGESAHAN	i
PERNYATAAN ORISINALITAS	i
KATA PENGANTAR.....	vi
ABSTRAK	vii
DAFTAR ISI	ix
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN	xiv
BAB 1 PENDAHULUAN.....	Error! Bookmark not defined.
1.1 Latar Belakang	Error! Bookmark not defined.
1.2 Rumusan Masalah	Error! Bookmark not defined.
1.3 Tujuan	Error! Bookmark not defined.
1.4 Manfaat	Error! Bookmark not defined.
1.5 Batasana Masalah.....	Error! Bookmark not defined.
1.6 Sistematika Pembahasan.....	Error! Bookmark not defined.
BAB 2 LANDASAN KEPUSTAKAAN	Error! Bookmark not defined.
2.1 Kajian Pustaka.....	Error! Bookmark not defined.
2.2 <i>Internet of Things (IoT)</i>	Error! Bookmark not defined.
2.3 pH Air	Error! Bookmark not defined.
2.5 MQTT (<i>Message Queuing Telemetry Transport</i>).....	Error! Bookmark not defined.
2.7 Mikrokontroler Wemos D1 (<i>Chipset ESP8266</i>).....	Error! Bookmark not defined.
2.8 Sensor pH Probe	Error! Bookmark not defined.
2.9 Mikrokomputer Raspberry Pi 2	Error! Bookmark not defined.
2.10 MongoDB	Error! Bookmark not defined.
2.11 Python.....	Error! Bookmark not defined.
2.12 Java	Error! Bookmark not defined.
BAB 3 METODOLOGI	Error! Bookmark not defined.
3.1 Studi Literatur	Error! Bookmark not defined.

3.2 Analisis Kebutuhan	Error! Bookmark not defined.
3.2.1 Analisis Kebutuhan Perangkat Keras.....	Error! Bookmark not defined.
3.2.2 Analisis Kebutuhan Perangkat Lunak.....	Error! Bookmark not defined.
3.2.3 Analisis Kebutuhan Fungsional	Error! Bookmark not defined.
3.3 Perancangan	Error! Bookmark not defined.
3.4 Implementasi	Error! Bookmark not defined.
3.5 Pengujian	Error! Bookmark not defined.
3.6 Kesimpulan	Error! Bookmark not defined.
BAB 4 REKAYASA KEBUTUHAN.....	Error! Bookmark not defined.
4.1 Deskripsi Umum.....	Error! Bookmark not defined.
4.1.1 Tujuan.....	Error! Bookmark not defined.
4.1.2 Ruang Lingkup	Error! Bookmark not defined.
4.1.3 Batasan Sistem	Error! Bookmark not defined.
4.1.4 Lingkungan Operasi.....	Error! Bookmark not defined.
4.2 Analisis Kebutuhan Sistem.....	Error! Bookmark not defined.
4.2.1 Kebutuhan Fungsional.....	Error! Bookmark not defined.
4.2.2 Kebutuhan Non-Fungsional.....	Error! Bookmark not defined.
BAB 5 PERANCANGAN DAN IMPLEMENTASI	Error! Bookmark not defined.
5.1 Perancangan Sistem	Error! Bookmark not defined.
5.1.1 Perancangan Sensor Node	Error! Bookmark not defined.
5.1.2 Perancangan Server Broker.....	Error! Bookmark not defined.
5.1.3 Perancangan User Subscriber	Error! Bookmark not defined.
5.1.4 Perancangan Database.....	Error! Bookmark not defined.
5.1.5 Lingkungan Pengujian	Error! Bookmark not defined.
5.2 Implementasi	Error! Bookmark not defined.
5.2.1 Implementasi Perangkat Keras Publisher.....	Error! Bookmark not defined.
5.2.2 Implementasi Perangkat Lunak Publisher.....	Error! Bookmark not defined.
Bab 6 PENGUJIAN.....	Error! Bookmark not defined.
5.1 Perancangan Pengujian	Error! Bookmark not defined.

5.1.1 Perancangan Pengujian Fungsional	Error! Bookmark not defined.
5.1.2 Perancangan Pengujian Kinerja Protokol	Error! Bookmark not defined.
5.2 Hasil Pengujian dan Analisis	Error! Bookmark not defined.
5.2.1 Hasil Pengujian dan Analisis Fungsional	Error! Bookmark not defined.
5.2.2 Hasil Pengujian dan Analisis Kinerja protokol	Error! Bookmark not defined.
BAB 7 KESIMPULAN	Error! Bookmark not defined.
7.1 Kesimpulan	Error! Bookmark not defined.
7.2 Saran	Error! Bookmark not defined.
DAFTAR PUSTAKA.....	Error! Bookmark not defined.



DAFTAR TABEL

Tabel 2.1 Hubungan pH air dengan budidaya ikan ...	Error! Bookmark not defined.
Tabel 2.2 Spesifikasi Wemos D1.....	Error! Bookmark not defined.
Tabel 2.3 Fungsi Pin Wemos D1	Error! Bookmark not defined.
Tabel 3.1 Tabel Uji <i>Middleware</i>	18
Tabel 3.2 Tabel Uji Lingkup Area.....	19
Tabel 4.1 Kebutuhan Fungsional	Error! Bookmark not defined.
Tabel 4.2 Spesifikasi Sensor Node.....	Error! Bookmark not defined.
Tabel 4.3 Spesifikasi Server <i>Middleware</i>	Error! Bookmark not defined.
Tabel 4.4 Spesifikasi User Subscriber (PC)	Error! Bookmark not defined.
Tabel 4.5 Kebutuhan Perangkat Lunak	Error! Bookmark not defined.
Tabel 5.1 Keterangan Pin Sensor	Error! Bookmark not defined.
Tabel 5.2 Implementasi Program Library	35
Tabel 5.3 Implementasi inisialisasi untuk menghubungkan dengan WiFi dan MQTT Server.....	35
Tabel 5.4 Implementasi inisialisasi untuk publish dengan data Json.....	Error! Bookmark not defined.
Tabel 5.5 Implementasi on/off koneksi 8 menit	36
Tabel 5.6 Update, Upgrade dan Install Python.....	36
Tabel 5.7 Script PubSub <i>Middleware</i>	37

DAFTAR GAMBAR

Gambar 2.1 Skala pH air	Error! Bookmark not defined.
Gambar 2.2 Gambaran umum sistem IoT dengan MQTT	Error! Bookmark not defined.
Gambar 2.3 Wemos D1 Board	Error! Bookmark not defined.
Gambar 2.4 Sensor pH Probe	Error! Bookmark not defined.
Gambar 2.5 Mikrokomputer Raspberry Pi 2 Model B	Error! Bookmark not defined.
Gambar 2.6 Gambar tabel Spesifikasi Raspberry Pi	Error! Bookmark not defined.
Gambar 3.1 Diagram Alir Penelitian	Error! Bookmark not defined.
Gambar 3.2 Gambaran Umum Sistem	Error! Bookmark not defined.
Gambar 3.3 Kebutuhan Perangkat Keras	Error! Bookmark not defined.
Gambar 3.4 Kebutuhan Perangkat Lunak	16
Gambar 3.5 Alur Perancangan Sistem	Error! Bookmark not defined.
Gambar 5.1 Gambaran Umum Sistem	Error! Bookmark not defined.
Gambar 5.2 Rangkaian Sensor Node	Error! Bookmark not defined.
Gambar 5.3 Alur Kerja Sensor Node Publisher	Error! Bookmark not defined.
Gambar 5.4 Alur Kerja <i>Real Time Live Data</i>	Error! Bookmark not defined.
Gambar 5.5 Alur Kerja <i>Subscriber Middleware to Database</i>	Error! Bookmark not defined.
Gambar 5.6 Alur Kerja <i>Pub_Middleware harian</i>	Error! Bookmark not defined.
Gambar 5.7 Alur Kerja <i>Pub_Middleware History</i>	29
Gambar 5.8 Alur Kerja User Subscriber	Error! Bookmark not defined.
Gambar 5.9 Skema Collection MongoDB	Error! Bookmark not defined.
Gambar 5.10a Pengujian Lingkungan 5 Meter Hari Ke 1	Error! Bookmark not defined.
Gambar 5.10b Pengujian Lingkungan 10 Meter Hari Ke 2	Error! Bookmark not defined.
Gambar 5.10c Pengujian Lingkungan 15 Meter Hari Ke 3	Error! Bookmark not defined.
Gambar 5.11 Implementasi Sensor Node Publisher	Error! Bookmark not defined.
Gambar 5.12 Implementasi Hasil node sensor	Error! Bookmark not defined.
Gambar 5.13 Pengaktifan <i>Broker.py</i> dengan skema tidak pada waktunya menyimpan	Error! Bookmark not defined.
Gambar 5.14 Pengaktifan <i>Broker.py</i> dengan skema pada waktunya menyimpan	Error! Bookmark not defined.
Gambar 6.1 Rancangan Pengujian Kinerja protokol	Error! Bookmark not defined.
Gambar 6.2 Rancangan Pengujian Cakupan Area	Error! Bookmark not defined.
Gambar 6.3 Grafik pengujian kinerja protokol dengan <i>Jmeter</i>	Error! Bookmark not defined.

Gambar 6.4 Grafik Pengujian Jarak Cakupan Area**Error! Bookmark not defined.**



DAFTAR LAMPIRAN

A.1 Script Arduino publisher.....**Error! Bookmark not defined.**2

A.2 Wawancara Pakar Di BPBAP Situbondo ..**Error! Bookmark not defined.**5





BAB 1 PENDAHULUAN

1.1 Latar Belakang

Pada tahun 2014 tingkat produksi budidaya ikan di Indonesia semakin meningkat yang menjadikan Indonesia sebagai negara kedua terbesar setelah China untuk produksi ikan berdasarkan data FAO 2016. Data tersebut dipublikasikan dalam sebuah forum Internasional oleh Kementerian Kelautan dan Perikanan (KKP) dan Masyarakat Akuakultur Indonesia (MAI) (Mufid, A., 2016). Namun, terdapat banyak tantangan yang sedang dihadapi saat ini dalam pengembangan akuakultur di Indonesia.

Banyak faktor yang terjadi dalam pembudidayaan ikan yaitu kondisi tempat penangkaran, kualitas atau kadar pH air yang digunakan, suhu air, serta pemberian makan dan kesesuaian makanan yang digunakan. Kesalahan para pembudidaya ikan merupakan faktor yang sering kali terjadi di Indonesia. Salah satu dari berbagai kesalahan tersebut adalah Air, dimana kualitas air yang digunakan dapat berdampak positif dan negatif pada keberlangsungan hidup ikan. Derajat pH air untuk budidaya ikan yang memenuhi syarat adalah 5 – 8,5 untuk air laut dan untuk budidaya ikan air tawar adalah 6,5 – 7,5 satuan pH (Nasa, 2015). Oleh karena itu untuk mengurangi tingkat kesalahan dalam pemantauan pH air dalam kolam ikan agar ikan tetap terjaga kesehatannya adalah dengan bantuan teknologi. Dalam sebuah wawancara mengatakan bahwa pH air sangat berpengaruh pada pembudidayaan ikan yang sedang dilakukan khususnya pada kolam tambak udang. Salah satu faktor perubahan pH adalah kondisi cuaca, misalnya terjadi hujan (Asdary, M., 2017). Sehingga pemantauan pH air diperlukan agar mengetahui kondisi sebenarnya pada saat itu. Kondisi cuaca yang tidak diharapkan kerap menjadi sebuah masalah para pengelola tambak untuk melakukan pemantauan pH. Sehingga untuk mengatasi masalah tersebut, para pembudidaya ikan memanfaatkan teknologi yang dapat membantu mereka dalam proses pemantauan pH air dengan segala kondisi. Salah satu pemanfaatan teknologi saat ini yang dapat digunakan adalah IoT atau *Internet of Things*.

Banyaknya perkembangan teknologi di era sekarang membuat banyak dampak perubahan bagi semua kalangan. Salah satu perkembangan teknologi saat ini ialah *Internet of Things* (IoT) yang merupakan sebuah penemuan yang saat ini banyak digunakan dan dialokasikan untuk segala aspek kehidupan. IoT sendiri merupakan sebuah gagasan dimana setiap benda-benda fisik yang terhubung dengan internet dalam satuan bentuk pemantauan dan dapat dikendalikan atau yang lainnya secara realtime (Alsaadi, E., 2015). Di Indonesia sendiri IoT telah berkembang dengan pesat banyaknya penelitian yang memanfaatkan sistem teknologi *Internet of Things*.

Dengan konsep tersebut penggabungan antara sebuah alat dengan sebuah jaringan nirkabel dapat digunakan pada sistem pengamatan ini. Sistem pengamatan ini memerlukan alat yang berfungsi sebagai kontrol yang berupa mikrokontroler, mikrokomputer dan akuisisi data berupa sensor yang saling terhubung dengan komunikasi jaringan wireless. Oleh karena itu, dibutuhkan sebuah protokol komunikasi yang dapat memadai. Protokol MQTT (*Message Queuing Telemetry Transport*) merupakan sebuah protokol dengan komunikasi yang ringan berbasis *publisher*, *broker*, dan *subscriber* dengan daya proses dan penyimpanan kecil (Aziz, B., 2016). MQTT merupakan protokol komunikasi yang dibangun untuk aplikasi *Machine-2-Machine* (M2M). Selain itu proses pertukaran pesan juga terjalin antara *publisher* dan

subscriber. Penggunaan protokol MQTT diharapkan dapat menciptakan suatu komunikasi data yang handal dan pemakaian *resource* yang rendah.

Dari pembahasan dan permasalahan yang didapat dapat disimpulkan bahwa, peneliti akan melakukan sesuatu implementasi untuk membantu proses kinerja dalam pembudidayaan ikan yang memfokuskan pada pH air. Dalam proses penelitian ini, peneliti mengambil peluang untuk memanfaatkan teknologi IoT, dengan bantuan protokol MQTT yang dapat digunakan dalam proses pengamatan pH Air, yang berfungsi sebagai sarana komunikasi antar *client* dan *server*. Dengan mekanisme *Publish* dan *Subscribe*, dimana terdapat node untuk sensor dengan mikrokontroler yang berfungsi sebagai *publisher*. *Publisher* akan mengirimkan hasil akuisisi kepada broker. Broker berperan sebagai kolektor data dengan bantuan komunikasi jaringan *access point* yang telah ditentukan. *Broker* ditempatkan pada mikrokomputer Raspberry Pi, yang memiliki fungsi untuk menyimpan hasil pengiriman data dari node sensor, dengan menggunakan bantuan database *mongoDB*. Proses penyimpanan data pada *mongoDB* akan diatur oleh *Pub_Sub_Middleware*. Terdapat *subscriber* yang bertugas untuk mendapatkan informasi pada *broker* sesuai topik yang sama dengan *interface*, yang juga sangat mudah dimengerti oleh pengguna.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan sebelumnya, maka rumusan masalah yang didapat yaitu sebagai berikut:

1. Bagaimana alur data dari perangkat *node sensor* sampai ke pengguna?
2. Bagaimana mekanisme yang terjadi didalam server?
3. Bagaimana menguji fungsionalisasi sistem dan kinerjansi pada protokol.

1.3 Tujuan

Adapun tujuan dari penelitian ini adalah :

1. Membangun sebuah sistem pengamatan yang berdasarkan dari sistem komunikasi dengan protokol *MQTT*.
2. Menganalisa kinerja protokol dengan melihat kinerjasi keseluruhan sistem.
3. Membantu pengguna yang menggunakan sistem ini khususnya pada bidang budidaya perikanan yang mengacu pada pentingnya pH yang sedang digunakan.

1.4 Manfaat

1. Memberikan kemudahan kepada pengguna, pemilik dan pengelola untuk mengontrol dan pemantauan kondisi pH air dari jarak jauh.
2. Mengimplementasikan konsep IoT dengan protokol MQTT pada sistem pemantauan.

1.5 Batasan Masalah

Ruang lingkup penelitian ini adalah perancangan dan penelitian sistem pengamatan berupa perangkat keras dan perangkat lunak. Adapun batasan penelitian ini antara lain :

1. Sistem saling berkomunikasi dengan jaringan wireless.
2. Jumlah *client publisher* berjumlah 2 dan *server* berjumlah 1.
3. Jumlah *client subscriber* bisa lebih dari satu.
4. *Broker* menggunakan *MQTT Mosquitto* pada mikrokomputer *Raspberry Pi*.

5. Interface *Subscriber* yang sederhana.

1.6 Sistematika Pembahasan

Penulisan, sistematika, pembahasan dan penyusunan laporan penelitian dapat diuraikan sebagai berikut:

BAB I Pendahuluan

Bab yang menjelaskan dibuatnya penelitian ini dengan uraian yang didukung oleh latar belakang, sehingga memperoleh rumusan masalah, manfaat dan batasan dalam penelitian tersebut.

BAB II Landasan Kepustakaan

Bab landasan kepastakan berisi tentang uraian dari studi literatur ataupun kajian pustaka yang diperoleh dari hasil penelitian-penelitian yang sudah pernah dilakukan. Uraian tersebut disajikan sebagai penunjang penelitian ini yang meliputi penerapan *IoT*, protokol MQTT, *Middleware* serta parameter uji yang digunakan.

BAB III Metodologi

Pada bab ini membahas beberapa langkah kerja yang akan dilakukan dalam penelitian, diantaranya studi literatur, analisis kebutuhan sistem, implementasi dan pengujian kinerja protokol MQTT pada sistem pengamatan pH air.

BAB IV Rekayasa Kebutuhan

Bab ini menjelaskan secara rinci mengenai deskripsi umum dari sistem, kebutuhan sistem yang meliputi kebutuhan perangkat keras perangkat lunak dan kebutuhan fungsional sistem.

BAB V Perancangan dan Implementasi

Membahas tentang perancangan sistem hingga hasil implementasi sistem yang berupa implementasi perangkat keras dan implementasi perangkat lunak.

BAB VI Pengujian

Bab ini berfungsi untuk melaporkan hasil pelaksanaan metode/teknik penelitian dan menyajikan data yang mendukung hasil tersebut. Penyajian data dan penjelasannya dilakukan secara terurut dan logis menggunakan teks dan ilustrasi lainnya (misalnya, tabel dan gambar).

BAB VII Penutup

Memuat kesimpulan serta saran yang diperoleh dari pembuatan dan pengujian metode untuk pengembangan lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Terdapat banyak penelitian yang telah dilakukan sebelumnya. Seperti yang berjudul "*A Publish/Subscribe Messaging System For Wireless Sensor Communication*" dimana, hal tersebut menjelaskan tentang keefisienan dalam menggunakan sistem Publish/Subscribe yang berperan sebagai sarana komunikasi untuk *Wireless Sensor Network* (WSN). Tetapi hal itu juga dikaitkan dengan harapan yang memiliki tujuan untuk menuntut protokol dapat menyediakan interkoneksi yang baik antar WSN yang berdasarkan pada teknologi berbeda tanpa memerlukan sebuah node untuk mengetahui dari mana data tersebut dihasilkan (Weiss, B. et al., 2016).

Pada penelitian sebelumnya, membahas tentang konsep IoT yang berjudul "*The Internet of Things A Survey*", penelitian tersebut menjelaskan mengenai sebuah fungsi umum yang disertai tujuan dan konsep komunikasi antar things atau sebuah benda yang saling terhubung dengan bantuan jaringan menggunakan komunikasi protokol. *IoT* terklasifikasi menjadi 3 bagian yaitu, yang pertama "*Things oriented*" hal ini berupa benda sederhana yang terdiri dari RFID, sensor dan akuator. Yang kedua "*Internet Oriented*", dimana hal ini merupakan things yang saling terhubung. Ketiga adalah "*Semantic Oriented*", bagian ini merupakan salah satu tantangan teknologi *IoT*, yang saling berkaitan dengan jumlah things yang terlibat dalam sebuah konsep pengaplikasian (Atzori, L. et al., 2015).

Penelitian sebelumnya telah menjabarkan pembahasan mengenai konsep komunikasi MQTT yang berjudul "*A formal model and analysis of an IoT Protokol*" ,yang didalamnya menjelaskan tentang fungsi ataupun penggunaan dari masing-masing konsep komunikasi MQTT yang berperan sebagai *client* dan *server*. Pada sistem ini juga menggunakan pengaturan yang terdapat alur komunikasi untuk saling bertukar pesan dengan acknowledgment (ACK) yang berperan sebagai pengirim atau produsen dari munculnya sebuah pesan, yang melibatkan fungsi yang memiliki kaitan atau sebuah satu kesatuan di dalamnya. (Aziz, B., 2016).

Berikutnya adalah penelitian yang berjudul "*Sistem of acquisition, transmission, storage and visualization of Pulse Oximeter and ECG data using Android and MQTT*". Penelitian tersebut menjelaskan tentang konsep komunikasi MQTT *publisher*, *broker*, dan *subscribe*. Pada protokol MQTT terdapat entitas perantara *client* yang bertindak sebagai *publish* maupun *subscribe* pada topik tertentu. Kemudian topik tersebut akan terdaftar pada *broker* dan sebagai *subscriber* mereka dapat berlangganan topik pada *broker* walaupun sebagai *subscriber* yang berbeda (Barata, D. et al., 2013).

Berdasarkan ringkasan penelitian sebelumnya penulis mendapatkan sebuah ide untuk mengimplementasikan protokol MQTT pada sebuah sistem pengamatan pH air menggunakan konsep *Internet of Things* (IoT). Sehingga nantinya dapat menganalisa kinerja pada protokol tersebut. Pada sistem yang dibuat akan menggunakan perantara mikrokontroler arduino WeMos D1 sebagai kontrol kendali dan sensor pH meter yang berfungsi sebagai pengambil data yang nantinya akan menjadi sensor node *publish*. Hasil yang didapat pada sensor node akan dikirimkan pada *server* atau yang disebut *broker* melalui komunikasi protokol MQTT yang saling terhubung dengan *request* dan *response* pada tindakan pertama. Selanjutnya hasil

data tersebut dapat dibagikan pada *client subscribe* yang melakukan *request* dengan topik yang telah ditentukan oleh *publish* pada *broker*.

2.2 Internet of Things (IoT)

Internet of Things (IoT) merupakan sebuah penemuan yang saat ini sangat banyak diminati oleh para pengguna yang memanfaatkan kombinasi benda-benda fisik dan internet. IoT juga tergolong signifikan karena dapat membuat sebuah objek melakukan aktifitas dirinya sendiri secara digital dan menjadi sesuatu yang lebih besar dari objek itu sendiri. IoT melibatkan perluasan konektivitas internet di luar perangkat standar seperti desktop, laptop, *smartphone*, dan tablet, dan keberbagai perangkat fisik atau benda sehari – hari secara manual atau *non-internet*. Dalam lingkup teknologi, perangkat ini dapat berkomunikasi dan berinteraksi melalui internet dan mereka dapat dimonitor serta dikendalikan dari jarak jauh. IoT dapat membuat berbagai macam proses menjadi lebih efisien, antara lain dalam hal penggunaan aset, produktivitas karyawan, *supply chain* dan logistik, pengalaman pelanggan, dan inovasi produk. Diperkirakan IoT memiliki dampak ekonomi potensial sebesar \$3.9-11.1 triliun per tahun pada tahun 2025 (Manyika, J. et al., 2015).

Arsitektur IoT terdiri dari jaringan, sistem yang kompleks dan juga keamanan yang ketat, jika ketiganya bisa dicapai maka kontrol otomatisasi dalam *Internet of Things* dan dapat berjalan dengan lancar. Komponen-komponen tersebut terdiri dari *identification*, *sensing*, *communication*, *computation*, *services* dan *semantics* (Mohammadi, m. et al., 2015). *Identification* pada IoT sangat diperlukan untuk membuat sebuah gagasan yang jelas bagi setiap objek yang akan dibuat. *Sensing* atau yang bisa dibilang sebagai pengolah data dari sebuah objek yang akan dikirim untuk *warehouse*, *cloud* maupun *database*. *Communication* adalah jembatan yang digunakan pada IoT untuk berinteraksi dalam sistem yang dapat memberikan layanan yang spesifik. *Computation* merupakan perangkat lunak yang mempresentasikan sebuah *processing* dalam komputasi IoT. Pada IoT *service* merupakan hal utama dalam memberikan layanan yang dapat diakses, pengumpulan data yang diperoleh, virtualisasi data fisik dan pengambilan keputusan dari data yang diperoleh.

2.3 pH Air

pH adalah singkatan dari pangkat Hydrogen yang merupakan tingkatan asam basa suatu larutan. Skala pH bukanlah skala absolut, ia bersifat relatif terhadap sekumpulan larutan standar yang PH-nya ditentukan berdasarkan persetujuan internasional. Air murni bersifat netral dengan pH-nya pada suhu 25°C yang ditetapkan sebagai 7,0. Larutan pH yang kurang disebut bersifat asam dan yang lebih disebut bersifat basa. Pengukuran pH sangatlah penting dalam bidang yang terkait dengan kehidupan atau industri. Berikut gambar 2.1 skala pH air.

4.0	5.0	6.0	6.6	7.0	7.4	7.6	8.5	9.8	10.0
Acidic				Netral		Alkaline			

Gambar 2.1 Skala pH air

Sumber : (Nasa, 2015)

pH air mempengaruhi tingkat asam basa perairan dan kehidupan jasad renik. Perairan asam akan kurang produktif dapat membunuh hewan budidaya. Pada pH rendah kandungan oksigen terlarut akan berkurang sebagai akibatnya konsumsi oksigen menurun, aktivitas pernafasan menurun, aktivitas pernafasan naik dan selera makan akan berkurang, hal sebaliknya terjadi pada suasana basa. Atas dasar ini maka usaha budidaya perairan akan berhasil baik dalam air dengan pH 6,5 – 9,0 dengan kisaran optimal 7,5 – 8,7.

Tabel 2.1 Hubungan pH air dengan budidaya ikan

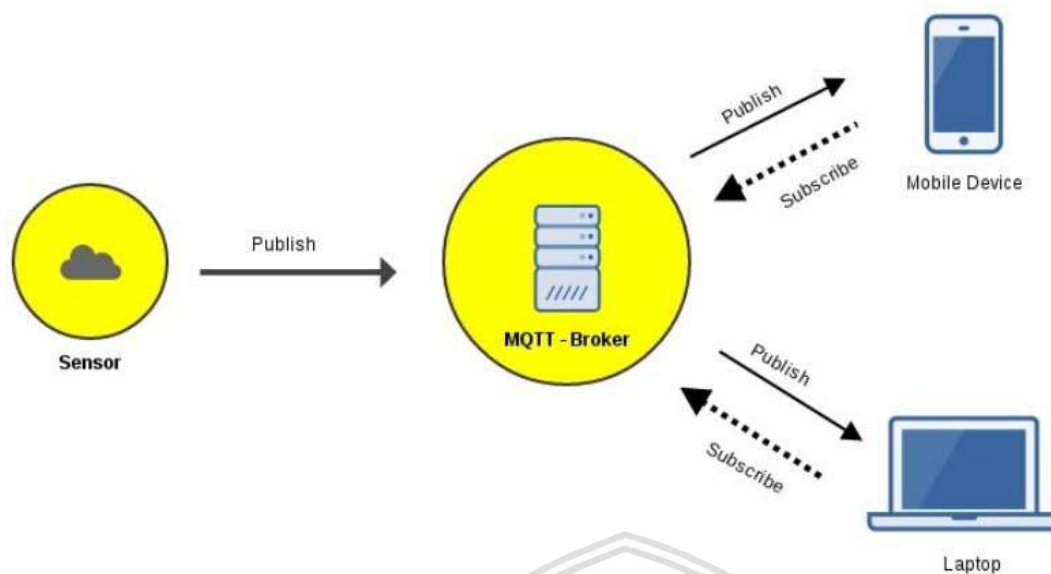
Kuaitas pH air	Pengaruh
< 4,5	Air bersifat racun bagi ikan
5 – 6,5	Pertumbuhan ikan terhambat dan mudah terjangkit bakteri dan parasite
6,5 – 9,0	Ikan mengalami pertumbuhan optimal dan baik
>9,0	Pertumbuhan ikan terhambat

Sumber : (Nasa, 2015)

2.5 MQTT (*Message Queuing Telemetry Transport*)

Protokol *Message Queuing Telemetry Transport* atau yang bisa disebut MQTT merupakan sebuah protokol komunikasi data M2M atau *Machine to Machine* yang berada pada layer aplikasi (Adi, H. K., et al., 2016). MQTT berkomunikasi dengan mengirimkan data pesan yang memiliki *header* berukuran kecil yaitu hanya sebesar 2 bytes untuk setiap jenis data, sehingga dapat bekerja di dalam lingkungan yang terbatas sumber daya seperti kecilnya *bandwidth* dan terbatasnya sumber daya listrik. protokol MQTT menggunakan metode *publish/subscribe* untuk metode komunikasi.

Publish/Subscribe sendiri adalah sebuah pola pertukaran pesan di dalam komunikasi jaringan dimana, pengirim data disebut dengan *publisher* dan penerima data disebut dengan *subscriber*. Metode *publish/subscribe* memiliki beberapa kelebihan, salah satunya yaitu "Loose Coupling" atau "Decouple" yang berarti, antara *Publisher* dan *Subscriber* tidak saling mengetahui keberadaannya. Terdapat 3 buah *Decoupling* yaitu *Time Decoupling*, *Space Decoupling*, dan *Synchronization Decoupling*. *Time decoupling* adalah sebuah kondisi dimana, *Publisher* dan *Subscriber* tidak harus saling aktif pada waktu yang sama, yang berarti memperbolehkan mereka berada dalam waktu yang bersamaan. *Space Decoupling* adalah dimana *publisher* dan *subscriber* aktif di waktu yang sama, yang berarti mewajibkan mereka untuk berinteraksi dengan waktu yang bersamaan, akan tetepi antara *publisher* dan *subscriber* tidak saling mengetahui keberadaan dan identitas satu sama lain. Terakhir adalah *Synchronization Decoupling* yaitu, kondisi dimana pengaturan event baik itu menerima atau mengirim pesan di sebuah node hingga tidak saling mengganggu satu sama lain. (Adi, et al., 2016).



Gambar 2.2 Gambaran umum sistem IoT dengan MQTT

Sumber : (Pr, 2015)

Pada setiap MQTT *publish* mengirim dengan salah satu dari tiga tingkatan. Tingkatan ini terkait dengan jaminan yang berbeda sehubungan pada kendala pengiriman pesan. Baik pada sisi *client* dan *broker* yang dapat memberikan tepatan dan meningkatkan kehandalan jika terjadi kegagalan dalam jaringan, *reload* aplikasi dan keadaan tidak terduga. MQTT mengandalkan *TCP*, yang memiliki kehandalan tersendiri, secara *history* tingkat *QoS* dibutuhkan untuk mengatasi kehilangan data pada jaringan *TCP* yang lebih lama dan tidak dapat diandalkan. Karena ini masih dalam taraf perhatian yang *valid* untuk sebuah jaringan seluler pada saat ini. Tiga tingkatan itu adalah *QoS level 0*, dimana *publisher* berdasarkan *IP-Based* melakukan *publish* kepada *server* atau *MQTT broker* akan diterima pada *subscriber*. Tingkatan *QoS level 1*, merupakan komunikasi yang terjadi diantara *publisher*, *broker*, dan *subscriber* yang memiliki jaminan bahwa data berhasil sampai pada tujuan, namun tidak memiliki jaminan data yang dikirim terduplikat. Tingkatan terakhir *QoS level 2*, yang memiliki jaminan bahwa pesan yang dikirim telah sampai pada *MQTT broker* dan dipastikan diterima oleh *MQTT subscriber*.

2.7 Mikrokontroler Wemos D1 (Chipset ESP8266)

Wemos D1 merupakan sebuah mikrokontroler pengembangan berbasis modul mikrokontroler ESP8266. Mikrokontroler ini dibuat sebagai alternatif lain karena mahalnya biaya yang digunakan untuk membangun sistem *WiFi* dengan menggunakan mikrokontroler Arduino dan *WiFi Shield*. Wemos diprogram menggunakan Arduino IDE dengan sintak dan *library* yang banyak terdapat di internet dan pin out yang *compatible* dengan Arduino Uno sehingga memudahkan untuk terhubung dengan Arduino *shield* lainnya (dewi, 2017). Mikrokontroler ini diproduksi oleh produsen Cina yang berbasis di Shanghai. Ai-Thinker membuat ESP-01 dengan menggunakan lisensi oleh *Espressif*. Modul ini memungkinkan mikrokontroler untuk terhubung dengan jaringan *WiFi* dan membuat koneksi *TCP/IP* hanya dengan menggunakan *command*. Dengan clock 80 MHz chip ini dibekali dengan 4MB eksternal RAM, mendukung format IEEE 802.11 b/g/n sehingga tidak menyebabkan gangguan bagi yang lain. Untuk spesifikasi Wemos D1 dapat dilihat pada tabel 2.2 (dewi, 2017).

Tabel 2.2 Spesifikasi Wemos D1

Mikrokontroler Wemos D1	
Operating Voltage	3.3V
Digital I/O Pins	11
Analog Input Pins	1(Max input: 3.2V)
Clock Speed	80MHz/160MHz
Flash	4M bytes
Length	34.2mm
Width	25.6mm
Weight	10g

Perangkat ini didesain atau dirancang dengan referensi seperti Arduino Uno sehingga pin – pin pada mikrokontroler ini memiliki fungsi yang mirip dengan pin – pin pada Arduino Uno. Oleh karena itu *mikrokontroler* ini dapat diprogram melalui Arduino IDE. Adapun fungsi – fungsi pin pada Wemos D1 dapat dilihat pada tabel 2.3.

Tabel 2.3 Fungsi Pin Wemos D1

Pin	Fungsi
TX	TXD
RX	RXD
A0	Analog Input, max 3.3V input
D0	IO
D1	IO, SCL
D2	IO, SDA
D3	IO, 10K Pull up
D4	IO, 10K Pull up, BUILTIN_LED
D5	IO, SCK
D6	IO, MISO
D7	IO, MOSI
D8	IO, 10K Pull up, SS
GND	Ground
5V	5V
3V3	3.3V
RST	Reset



Gambar 2.3 Wemos D1 Board

2.8 Sensor pH Probe

Sensor adalah sesuatu yang digunakan untuk mendeteksi adanya perubahan lingkungan fisik atau kimia. Variabel keluaran dari sensor yang diubah menjadi besaran listrik. Sensor pH Probe merupakan suatu sensor yang digunakan untuk mendeteksi kualitas pH terhadap cairan. Alat tersebut dapat digunakan berulang-ulang dilaboratorium dan lingkungan yang mau diukur kadar pHnya. Sensor tersebut merupakan peralatan yang banyak digunakan dalam pengukuran pH dalam air, tangka ikan, kolam renang, lab dan penggunaan lapangan, makanan yang aman, pengembangan fotografi, bir, anggur dan minuman keras lainnya. Sensor tersebut bisa membaca kadar pH dari rentang 0 s/d 14 yang berkualitas tinggi. Konektor yang cocok pada kebanyakan pH meter dan mikrokontroler. Ketahanan alat yang dapat bertahan lama (DFRobot, 2013). Adapun sensor pH probe dapat dilihat pada gambar 2.4 dan spesifikasi sensor pada tabel 2.4.



Gambar 2.4 Sensor pH Probe

Tabel 2.4 Spesifikasi pH Probe

Spesifikasi	
Range Pengukuran	0 – 14 pH
Resolusi	0,01
Akurasi	+ - 0,05
Range Temperatur	0°~50° C (32°F~122°F)
Dimensi	150 mm
Range Cakupan	±5 m

2.9 Mikrokomputer Raspberry Pi 2

Raspberry Pi 2 atau yang sering disebut dengan raspi, adalah sebuah mini komputer yang berukuran kecil seperti ukuran kartu kredit. Perangkat ini juga digunakan untuk menjalankan program perkantoran, permainan komputer, dan sebagai pemutar media hingga video beresolusi tinggi. Raspberry Pi 2 dikembangkan oleh yayasan Nirlaba, Raspberry Pi *Foundation*, yang dikontrol oleh sejumlah pengembang dan ahli komputer dari Universitas Cambridge, Inggris. Berikut gambar sebuah Raspberry Pi pada gambar 2.5.



Gambar 2.6 Mikrokomputer Raspberry Pi 2 Model B

Raspberry Pi 2 memiliki dua model: model A dan model B. Secara umum Raspberry Pi Model B memiliki kapasitas penyimpanan RAM sebesar 512 MB. Perbedaan model A dan B terletak pada modul penyimpanan yang digunakan. Model A menggunakan penyimpanan sebesar 256 MB dan penyimpanan model B sebesar 512 MB. Selain itu, model B sudah dilengkapi dengan porta Ethernet (untuk LAN) yang tidak terdapat di model A. Desain Raspberry Pi didasarkan pada SoC (*system-on-a-chip*) Broadcom BCM2835, yang telah menanamkan prosesor ARM1176JZF-S dengan 700 MHz, GPU VideoCore IV, dan RAM sebesar 256 MB (model B). Penyimpanan data tidak didesain untuk menggunakan cakram keras atau *solid-state drive*, melainkan mengandalkan kartu penyimpanan tipe SD untuk menjalankan sistem dan sebagai media penyimpanan jangka panjang. Berikut gambar spesifikasi Raspberry Pi pada gambar 2.6.

	Raspberry Pi 3 Model B	Raspberry Pi Zero	Raspberry Pi 2 Model B	Raspberry Pi Model B+
Introduction Date	2/29/2016	1/12/2015	2/2/2015	7/14/2014
SoC	BCM2837	BCM2835	BCM2835	BCM2835
CPU	Quad Cortex A53 @ 1.2GHz	ARM11 @ 1GHz	Quad Cortex A7 @ 900MHz	ARM11 @ 700MHz
Instruction set	ARMv8-A	ARMv6	ARMv7-A	ARMv6
GPU	400MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV
RAM	1GB SDRAM	512 MB SDRAM	1GB SDRAM	512MB SDRAM
Storage	micro-SD	micro-SD	micro-SD	micro-SD
Ethernet	10/100	none	10/100	10/100
Wireless	802.11n / Bluetooth 4.0	none	none	none
Video Output	HDMI / Composite	HDMI / Composite	HDMI / Composite	HDMI / Composite
Audio Output	HDMI / Headphone	HDMI	HDMI / Headphone	HDMI / Headphone
GPIO	40	40	40	40
Price	\$35	\$5	\$35	\$35

Gambar 2.6 Gambar tabel Spesifikasi Raspberry Pi

2.10 MongoDB

MongoDB adalah sebuah sistem basis data berorientasi dokumen yang *powerful*, fleksibel dan *scalable*. Sistem basis data ini menggantikan konsep "baris" menjadi sebuah "dokumen". Tidak seperti sistem basis data berorientasi "baris" yang menggunakan skema tetap dan didefinisikan diawal, sistem basis data dokumen tidak memiliki skema yang tetap.

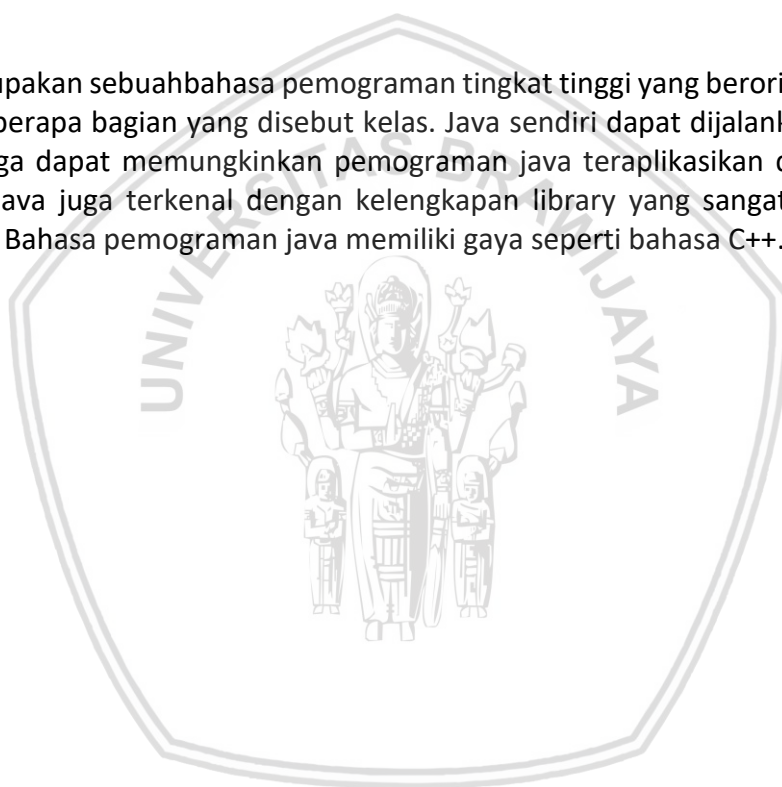
Sehingga hal tersebut menjadikan kunci dan nilai dari dokumen dapat berbeda ukuran maupun jenis. Dokumen adalah model data bagi mongoDB yang kurang lebih seperti satu baris dalam sistem basis data relasional. MognoDB menyimpan datanya dalam bentuk BSON. BSON adalah sebuah representasi biner dari dokumen JSON yang memiliki lebih banyak tipe data (Chodorow & Dirolf, 2010).

2.11 Python

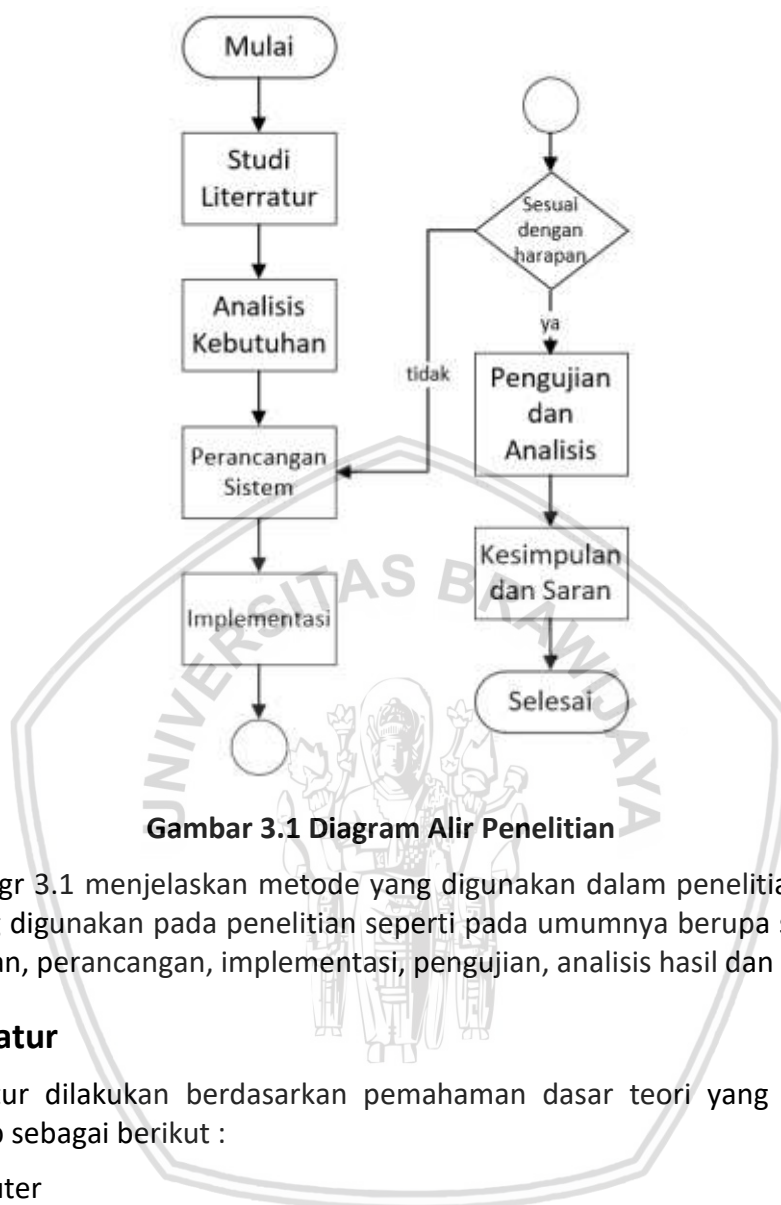
Python adalah bahasa pemrograman interpretatif multiguna. Tidak seperti bahasa lain yang susah untuk dibaca dan dipahami, Python lebih menekankan pada keterbacaan kode agar lebih mudah untuk memahami sintaks (Belajarpython, 2017). Untuk memperoleh hasil program yang sama, *source code* Python juga jauh lebih sedikit dibandingkan dengan kode yang ditulis menggunakan bahasa-bahasa pemrograman lain seperti C, C++, C# maupun Java.

2.12 Java

Java merupakan sebuah bahasa pemrograman tingkat tinggi yang berorientasi objek dan tersusun dari beberapa bagian yang disebut kelas. Java sendiri dapat dijalankan di beberapa *platform*, sehingga dapat memungkinkan pemrograman java teraplikasikan dalam berbagai sistem operasi. Java juga terkenal dengan kelengkapan library yang sangat memudahkan untuk pengguna. Bahasa pemrograman java memiliki gaya seperti bahasa C++.



BAB 3 METODOLOGI



Gambar 3.1 Diagram Alir Penelitian

Pada gambar 3.1 menjelaskan metode yang digunakan dalam penelitian pada sistem. Metodologi yang digunakan pada penelitian seperti pada umumnya berupa studi literature, analisis kebutuhan, perancangan, implementasi, pengujian, analisis hasil dan kesimpulan.

3.1 Studi Literatur

Studi literatur dilakukan berdasarkan pemahaman dasar teori yang terkait dengan beberapa konsep sebagai berikut :

1. Mikrokomputer
 - a) Raspberry Pi 2 model B.
2. Mikrokontroler
 - a) Wemos D1.
 - b) Konfigurasi sensor pH Meter.
Pada bagian ini melakukan kajian literatur bagaimana konfigurasi baca sensor serta mekanisme penyampaiannya.
3. Protokol MQTT
 - a) MQTT Server
Pada bagian ini melakukan kajian literatur tentang sistem bagian server.
 - b) MQTT Client
Pada bagian ini melakukan kajian literatur tentang komunikasi *arsitektur Publish Subscribe* dan cara kerjanya.
4. Database
 - a) MongoDB

Pada bagian ini melakukan kajian literatur tentang sistem basis data mongoDB dan cara menggunakannya.

5. Bahasa Pemrograman

a) Python

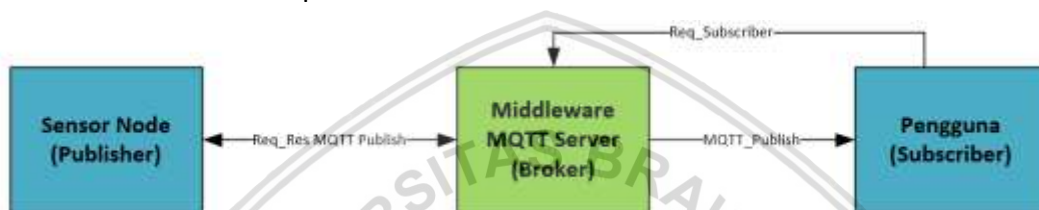
Pada bagian ini melakukan kajian tentang pemrograman dengan bahasa *python*.

b) Java

Pada bagian ini melakukan kajian tentang pemrograman dengan bahasa *java*.

3.2 Analisis Kebutuhan

Analisa kebutuhan ditujukan untuk melakukan analisis yang diperlukan sistem yang meliputi kebutuhan perangkat keras, perangkat lunak dan kebutuhan fungsional yang dapat dilakukan oleh sistem dalam penelitian ini.

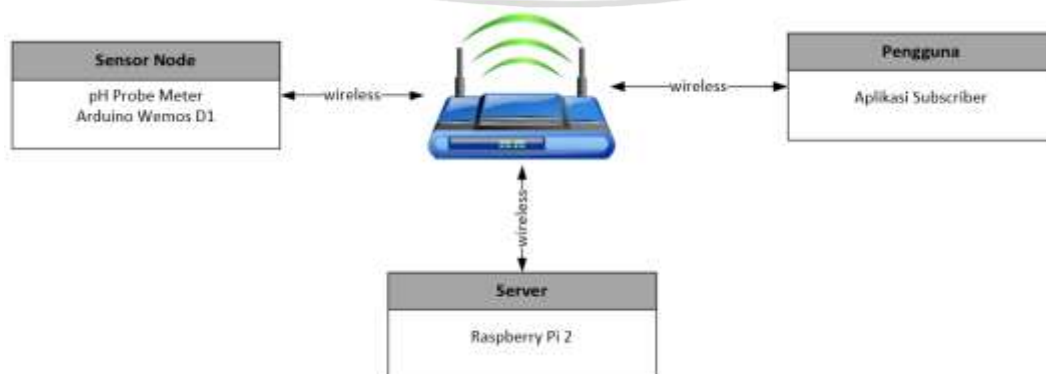


Gambar 3.2 Gambaran Umum Sistem

Pada gambar 3.2 terdapat gambaran umum sistem yang terdiri dari 3 entitas utama yang dirancang untuk membangun sistem yaitu *Sensor Node*, *Middleware*, *Subscriber*. Pada bagian *sensor node* merupakan bagian *publisher* yang berupa sensor dan mikrokontroler yang bekerja untuk mengirim data yang didapat dari objek yang diambil. *Middleware* merupakan bagian *Server Broker* yang berupa *mikrokomputer* yang mencakup *MQTT Broker*, *Database MongoDB*, *PubSub Middleware*. *Subscriber* merupakan bagian yang digunakan oleh pengguna untuk melihat hasil dari data yang didapatkan *sensor node* berupa *aplikasi subscriber* dengan bahasa pemrograman *java*.

3.2.1 Analisis Kebutuhan Perangkat Keras

Untuk mewujudkan sistem dalam penelitian ini, dibutuhkan analisis kebutuhan perangkat keras yang dapat digunakan, yaitu:



Gambar 3.3 Kebutuhan Perangkat Keras

Seperti yang telah digambarkan pada Gambar 3.3 kebutuhan perangkat keras yang digunakan dalam penelitian ini dibagi menjadi 3 bagian:

1. Sensor Node
 - a. *pH Probe Meter* berfungsi sebagai alat yang berhubungan langsung dengan objek yaitu air.
 - b. *Arduino Wemos D1* berfungsi sebagai *mikrokontroler* yang mengatur keseluruhan aktivitas dari sensor.
2. Pengguna
 - a. *Aplikasi Subscriber* berfungsi untuk jendela *interface* sistem dengan user yang menggunakan perangkat *device* seperti PC.
3. Middleware
 - a. Raspberry Pi 2 berfungsi sebagai wadah dari *Server Broker* beserta cakupan yang digunakan dalam *Server Broker*.

3.2.2 Analisis Kebutuhan Perangkat Lunak

Dalam penelitian ini diperlukan analisis kebutuhan perangkat lunak. Perangkat lunak tersebut akan diimplementasikan pada *hardware* atau perangkat keras sesuai dengan tujuannya masing-masing. Adapun perangkat lunak yang diperlukan dalam penelitian ini, yaitu :



Gambar 3.4 Kebutuhan Perangkat Lunak

Seperti yang telah digambarkan pada Gambar 3.4 kebutuhan perangkat lunak dalam penelitian ini dibagi menjadi 3 bagian:

1. Sensor Node
 - a. *Sensor Library* merupakan *library source code* yang dikhususkan untuk sebuah sensor untuk membaca objek. Berbeda sensor maka *source codenya* juga berbeda.
 - b. *MQTT Library publisher* merupakan *library source code* yang dikhususkan untuk *mikrokontroler* agar dapat mengolah *code* sensor dan melakukan kelanjutannya yaitu *publish data*.
 - c. *Json Library* merupakan *library source code* digunakan untuk pemformatan data yang telah didapat oleh sensor agar dapat disimpan kedalam mongoDB.
2. Pengguna (*Subscriber*)
 - a. *Java* merupakan bahasa pemrograman yang digunakan dalam penelitian ini untuk membuat sebuah *interface subscriber* bagi user.
 - b. *MQTT Library Subscriber* merupakan *library source code* yang digunakan untuk meng *subscribe* data dari *server* agar *pengguna* dapat mengetahui hasil dari pembacaan sensor.
3. *Middleware*

- a. *MQTT Broker* berfungsi sebagai pelantara antara *Publisher* dan *Subscriber* dalam komunikasi data yang dilakukan oleh *publish/subscribe*.
- b. *PubSub Middleware* merupakan sebuah fitur yang memiliki 2 fungsi yaitu *Pub Middleware* data dari *MongoDB* untuk *pengguna subscriber* dan *Sub Middleware* yang dilakukan oleh sensor kemudian menyimpannya di *MongoDB*.
- c. *MongoDB Database* berfungsi sebagai database yang digunakan untuk menyimpan data yang dilakukan oleh *Sub Middleware*.

3.2.3 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional pada sistem ini merupakan tujuan atau fungsi layanan yang harus disesuaikan oleh sistem, antara lain:

1. Sensor Node mampu membaca objek kemudian mengolahnya dan mempublish data tersebut dengan topik, *authentikasi*, dan identitas pesan data yang dikirim.
2. Server dapat menerima koneksi dari *client publisher* ataupun *client subscriber* yang memiliki *authentikasi* yang benar.
3. Sistem dapat menyimpan data *publish* dari sensor dengan waktu yang ditentukan.
4. Sistem dapat menampilkan hasil data *real time* dari sensor dan menampilkan data yang diinginkan di luar waktu *realtime*.

3.3 Perancangan

Perancangan sistem pada tahap ini dilakukan agar proses pembuatan terlaksana dengan terarah dan terstruktur. Perancangan dilakukan setelah memenuhi semua aspek kebutuhan yang meliputi kebutuhan perangkat lunak, perangkat keras, dan fungsional. Berikut alur rancangan yang akan dilakukan pada penelitian ini.



Gambar 3.5 Alur Perancangan Sistem

Berdasarkan gambar 3.5 terdapat 6 *statmen* alur rancangan yang dilakukan dalam penelitian ini.

3.4 Implementasi

Pada tahapan implementasi sistem ini dilakukan dengan melihat langkah tahapan yang telah dilakukan sebelumnya yaitu pada tahap analisis kebutuhan dan perancangan. Berikut implementasi yang akan dilakukan pada sistem ini:

1. Implementasi *Server Broker* yang menjelaskan tentang instalasi *Mosquitto* sebagai *MQTT Broker* dan *Authentikasi penggunaan password*.
2. Implementasi *Pub_Sub Middleware* yang menjelaskan tentang proses *Publish Subscribe* data yang disimpan pada *MongoDB*.
3. Implementasi *Database* yang menjelaskan tentang instalasi *MongoDb*.
4. Implementasi *Sensor Node* yang menjelaskan tentang *Source Code* yang digunakan *Publisher*.
5. Implementasi *subscriber* pengguna yang menjelaskan tentang *Source Code* yang digunakan pada *Subscriber*.

3.5 Pengujian

Dalam penelitian ini tahap pengujian akan dilakukan dengan beberapa skenario pengujian yang diharapkan dapat menghasilkan pengujian yang nantinya akan dilakukan analisis terhadap hasil pengujian tersebut. Adapun beberapa skenario tersebut adalah sebagai berikut :

1. Pengujian dengan menjalankan keseluruhan sistem atau pengujian fungsionalitas yang bertujuan untuk melihat apakah sistem sesuai dengan apa yang dirancang. Dengan beberapa skenario pengujian dari fitur-fitur yang dibuat. Berikut skenario pengujian fungsionalitas yang digunakan.
 - a) Melihat pengiriman data dari sensor node ke *server*.
 - b) Melihat pengiriman data *PubSub* dari *middleware* ke user.
 - c) Melihat kesesuaian *aplikasi* user sebagai *subscriber*
2. Pengujian kinerja dari protokol dengan menggunakan beberapa skenario yang dilakukan. Berikut skenario pengujian kinerja.
 - a) Pengujian *middleware* menerima data *sensor* secara langsung

Pengujian dilakukan dengan melihat dari *kinerjansi server* dengan melihat berapa banyak *middleware* mampu mengatasi pengiriman data dari *sensor node*. Dengan mengacu pada 2 parameter uji yaitu *latency* dan *error rate*. Banyaknya skenario uji yang digunakan adalah 100, 200, 300 pengguna melakukan pengiriman secara bersamaan. Berikut tabel uji kinerja *middleware*.

Tabel 3.1 Tabel Uji Middleware

No	Jumlah pengguna	<i>Latency</i>	<i>Error rate</i>
1	100	-	-

2	200		
3	300	-	-

b) Pengujian dalam cakupan area

Pengujian ini akan dilakukan dengan penempatan jarak teraman yang digunakan untuk lingkup AP (*Access Point*) yang dimana protokol MQTT diketahui adalah sebuah protokol yang digunakan pada perangkat terbatas dan lemah sinyal. Dengan mengacu pada parameter *delay* peneliti menggunakan skenario uji jarak 5, 10, 15 meter. Setiap jarak aja dilakukan dalam waktu 24 jam. Berikut tabel pengujian akurasi pengiriman.

Tabel 3.2 Tabel Uji lingkup area

No	Jarak ke AP (<i>Access Point</i>)	<i>Delay</i>
1	5 meter	-
2	10 meter	-
3	15 meter	-

Pada pengujian cakupan area sensor yang menjadi *publisher* untuk menyampaikan hasil data yang diperoleh hanya melakukan pengiriman 1 kali dalam waktu 8menit. Jadi dalam kurun waktu 8 menit sensor node akan aktif untuk memuai koneksi kepada server. Setelah terhubung dengan *server*, sensor akan melakukan *publish*. Kemudian setelah berhasil melakukan *publish*, *sensor node* akan memutuskan koneksi pada *server*. Pengulangan akan dilakukan kembali setelah 8 menit dari sensor terputus dari *server*.

3.6 Kesimpulan

Pada tahap ini pengambilan kesimpulan dilakukan setelah semua tahapan prancangan, implementasi dan pengujian telah selesai dilakukan. Kesimpulan diambil dari hasil analisis yang telah dilakukan. Tahap terakhir adalah saran yang dimaksudkan untuk memperbaiki kesalahan yang terjadi dalam penelitian dan menyempurnakan penulis serta memberikan pertimbangan atas pengembangan selanjutnya.

BAB 4 REKAYASA KEBUTUHAN

4.1 Deskripsi Umum

Pada bab ini menjelaskan tentang hal-hal yang harus terpenuhi dalam melakukan perancangan sampai implementasi dengan harapan penelitian ini dapat berjalan dengan lancar dan baik.

4.1.1 Tujuan

Tujuan dari bab ini adalah menjelaskan kebutuhan pembangunan sistem dari penelitian secara rinci dan detail. Menjelaskan sistem pengamatan dengan metode komunikasi MQTT *publish/subscribe* dan jaringan sensor berdasarkan *Software Requirement Specification (SRS) IEEE 830*. Dimulai dari komponen yang dibutuhkan dalam pelaksanaan perancangan serta bagaimana sistem akan bekerja dan beroperasi. Sehingga mendapatkan dokumentasi yang jelas mengenai alur kerja, pembuatan dan tujuan. Pembangunan sistem yang ditujukan untuk stakeholder atau pemangku kepentingan dan para pengembang sistem selanjutnya.

4.1.2 Ruang Lingkup

Dalam rancang bangun sistem ini untuk mengetahui bagaimana penerapan konsep komunikasi protokol MQTT. *Pub_Sub_Middleware, Database*, perancangan *client SensorNode Publisher* dan *client Subscriber* pengguna yang telah dilakukan dapat saling bertukar data. Pembuatan sistem ini ditujukan kepada penguji, developer dan pengguna sehingga kedepannya penelitian ini dapat dikembangkan lebih lanjut dan diharapkan dapat dimanfaatkan guna keperluan industri yang berlingkup dibidang yang sama maupun dikombinasikan dengan lainnya.

4.1.3 Batasan Sistem

Berikut batasan sistem yang digunakan:

1. Sistem akan bekerja apabila terkoneksi dengan satu jaringan yang sama.
2. Sistem beroperasi hanya berkomunikasi dalam jaringan *wireless*.
3. Sistem bekerja dengan sensor node dan subscriber yang telah dirancang dan dapat melakukan komunikasi dengan *Server Broker*.

4.1.4 Lingkungan Operasi

Berikut lingkungan operasi yang digunakan:

1. Terdapat air yang digunakan sebagai objek penelitian sistem.
2. Suhu kondisi air tidak boleh lebih dari 30 °C.
3. Kondisi Sensor harus dicelupkan kedalam air agar mendapatkan hasil yang akurat dengan rancangan *realtime*.

4.2 Analisis Kebutuhan Sistem

Pada bagian ini menjelaskan keseluruhan yang diperlukan oleh sistem agar dapat bekerja sesuai dengan tujuan yang telah diperoleh. Proses analisis dibagi menjadi dua yaitu kebutuhan fungsional dan kebutuhan non-fungsional.

4.2.1 Kebutuhan Fungsional

Pada tahap kebutuhan fungsional sistem harus mampu menjalankan skema yang telah dilakukan agar sistem dapat dinyatakan sesuai kebutuhan. Berikut adalah kebutuhan fungsional dari sistem.

Tabel 4.1 Kebutuhan Fungsional

No	Fitur
1.	Sistem dapat membaca objek dan mengolahnya untuk dipublish dengan <i>topik</i> , <i>autentikasi</i> , dan <i>identitas pesan data</i> yang dikirim dengan translasi ke Json.
2.	Sistem dapat menyimpan data <i>Publish</i> dari sensor dengan waktu yang ditentukan serta melakukan penginformasian dari data yang telah disimpan kepada <i>subscriber</i> .
3.	Sistem dapat menampilkan hasil data <i>realtime</i> dan data diluar waktu <i>realtime</i> .

4.2.2 Kebutuhan Non-Fungsional

Kebutuhan tahap ini berupa alat pendukung dalam penelitian. Kebutuhan non-fungsional yang dapat bagi menjadi dua bagian yaitu kebutuhan perangkat keras dan kebutuhan perangkat lunak.

4.2.2.1 Kebutuhan Perangkat Keras

Seperti yang telah dijelaskan pada Gambar 3.3 adapun kebutuhan perangkat keras yang dibutuhkan menjadi tiga bagian yaitu Sensor Node, Server Middleware, dan Subscriber pada device PC.

Tabel 4.2 Spesifikasi Sensor Node

Komponene	Spesifikasi
<i>pH Probe Meter</i>	Daya : 5 v Skala (pH) : 0-14 pH Temperatur : 0-50°C Akurasi : ± 0.1 pH
<i>Arduino Wemos D1</i>	CPU : ESP-8266EX Clock Speed : 80MHz/160MHz Flash : 4M byte

Tabel 4.3 Spesifikasi Server Middleware

Komponen	Spesifikasi
<i>Raspberry py 2 model B</i>	CPU : Quad cortex A7 RAM : 1GB Storage : 8GB Sistem Operasi : Raspbian Jessie

Tabel 4.4 Spesifikasi Subscriber (PC)

Komponen	Spesifikasi
<i>Processor</i>	<i>Intel(R)Core(TM) i7-7500U</i>
<i>Memory (RAM)</i>	<i>8GB DDR3</i>
<i>Hardisk</i>	<i>1000GB</i>
<i>Sistem operasi</i>	<i>Windows 10 Enterprise</i>

4.2.2.2 Kebutuhan Perangkat Lunak

Seperti yang digambarkan pada Gambar 3.4 kebutuhan perangkat lunak pada sistem juga terbagi menjadi tiga bagian yakni *SensorNode*, *Server Middleware* dan *Subscriber*. Adapun piranti yang mencakupi ketiganya sebagai berikut.

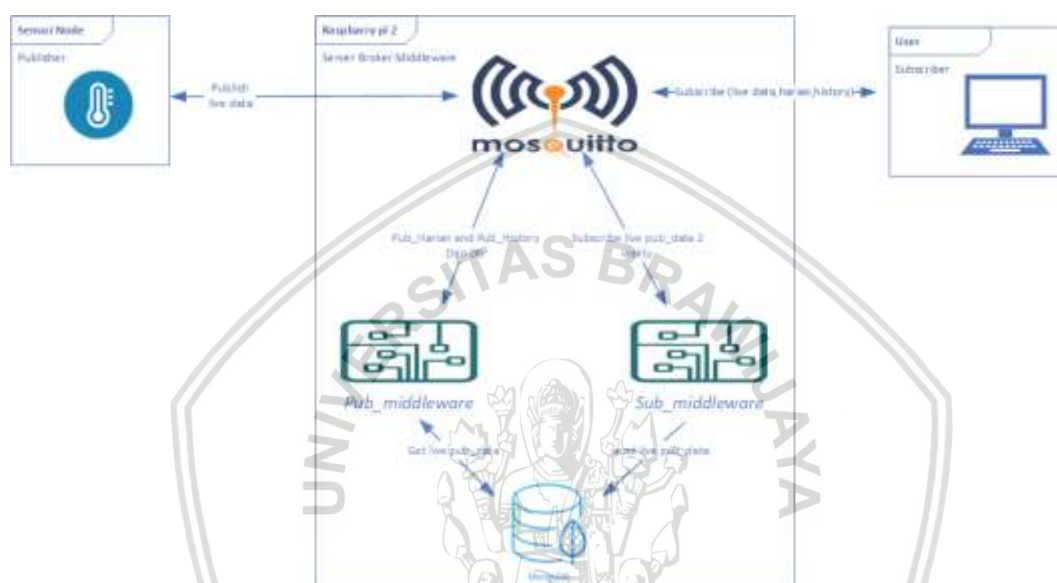
Tabel 4.5 Kebutuhan Perangkat Lunak

SensorNode	Server Middleware	Pengguna (Subscriber)
1. <i>Arduino Ide</i>	1. <i>Mosquitto</i> 2. <i>Python</i> 3. <i>MongoDb</i>	1. <i>Netbean (java)</i>

BAB 5 PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan Sistem

Pada bab ini menjelaskan tentang detail tahapan perancangan sistem pada penelitian ini yang dibangun mulai dari perancangan perangkat keras, perangkat lunak dan pendukung lainnya. Sistem yang dibuat meliputi beberapa komponen penting dan akan menjelaskan bagaimana alur kerja yang terdiri dari perancangan sensor node, perancangan *broker middleware*, perancangan *database*, dan perancangan *subscriber*.

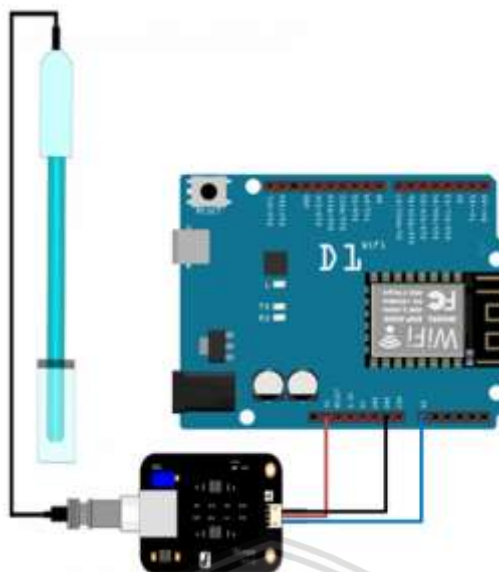


Gambar 5.1 Gambaran Umum Sistem

Seperti yang telah di gambarkan pada Gambar 5.1 perancangan sistem terdiri dari beberapa komponen penting yang saling berhubungan.

5.1.1 Perancangan Sensor Node

Sensor node merupakan bagian terpenting yang berfungsi sebagai pengambilan data atau pengamatan terhadap kondisi pH air. Dimana sensor node tersebut terdiri dari Mikrokontroler *WeMos D1* dan Sensor *pH Meter*. Rancangan sensor node dapat dilihat seperti Gambar 5.2.



Gambar 5.2 Rangkaian Sensor Node

Pada gambar 5.1 rangkaian pin – pin antara sensor dengan mikrokontroler saling terhubung menggunakan kabel jumper. Pada sensor pH Meter terdapat 3 pin yang dapat digunakan yaitu VCC(+), A0 (analog), GRD (-). Tabel 5.1 akan memperjelas hubungan antara pin sensor dengan mikrokontroler yang ditranslasi dengan modul pengkondisi sensor pH.

Tabel 5.1 Keterangan Pin Sensor

Modul Pengkondisi pH	Mikrokontroler Wemos D1
VCC	5V
GND	GND
A0	A0

Tabel 5.2 Spesifikasi Modul Pengkondisi pH

Spesifikasi	
Vin (arus)	DC (5V)
Range pengukuran	0 – 14 pH
Range Temperatur	-10 ~ 50 °C
Respon Time	5 s
Stabilization Time	60 s
Output	Analog
Dimensi	42 mm x 32 mm x 15 mm

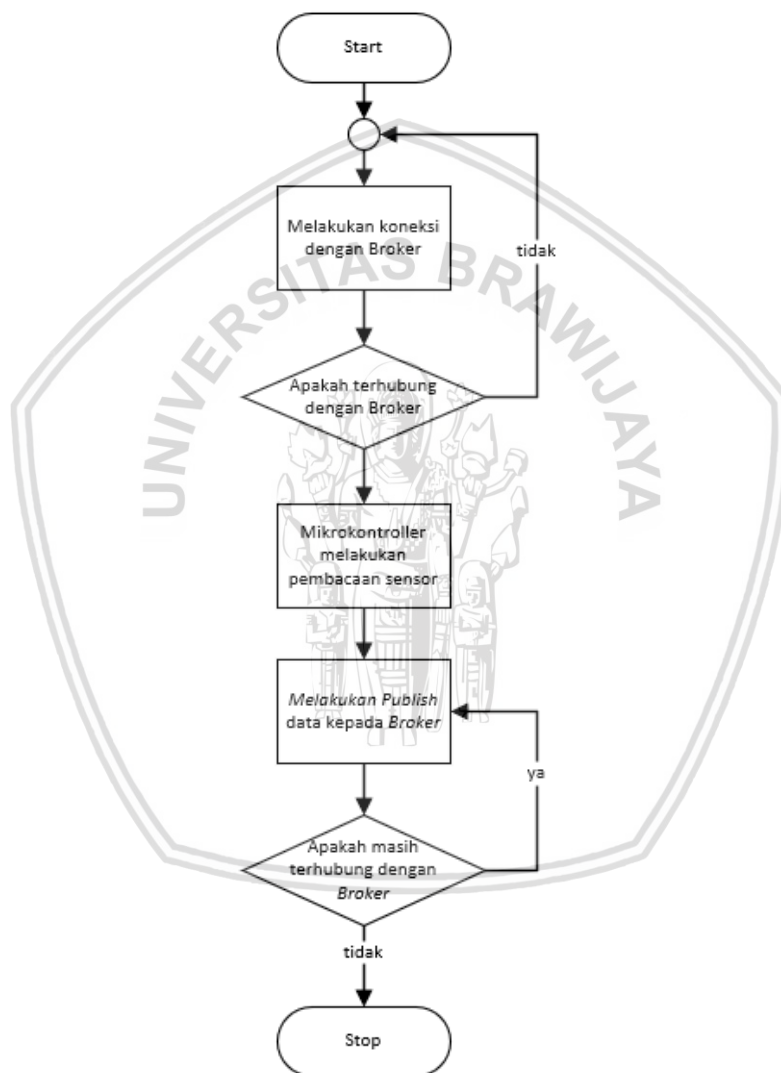
Pada tabel 5.2 merupakan keterangan spesifikasi modul translasi sensor pH dengan mikrokontroler arduino.

5.1.1.1 Alur Kerja Sensor Node

Sensor node berfungsi sebagai komponen *publisher* yang mengambil data pH langsung yang nantinya akan dikirim kepada user. Komponen sensor node bekerja dengan konfigurasi yang telah dirancang dan disematkan *code* yang berada di *mikrokontroler Arduino*. Berikut

cara kerja dan alur kerja dari sensor node *publisher* seperti yang ditunjukkan pada Gambar 5.3.

1. *Mikrokontroler* meminta *request connect* kepada *server broker* dengan membawa *key autentikasi* agar terkoneksi dan saling terhubung.
2. Setelah terhubung *mikrokontroler* akan melakukan pembacaan sensor dan mengolah hasil datanya.
3. Mengirimkan hasil pembacaan sensor dan mengirimkan kepada *server* dengan cara mempublish data tersebut.
4. Jika masih terhubung *mikrokontroler* akan melakukan *looping*.



Gambar 5.3 Alur Kerja Sensor Node Publisher

5.1.2 Perancangan Server Broker

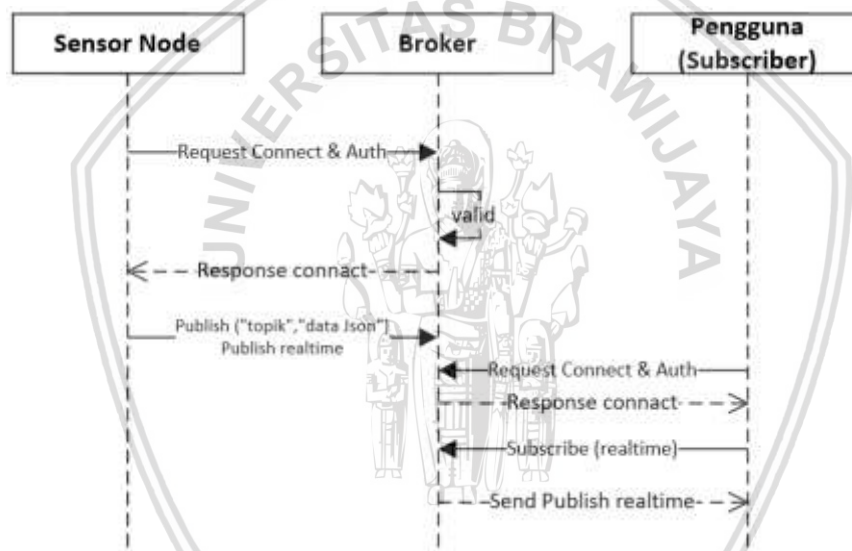
Pada *Server Broker Middleware*, perancangan perangkat keras menggunakan komponen *mikrokomputer* Raspberry Pi 2 model B. Sedangkan perancangan perangkat lunak menggunakan sistem operasi *Rasbian Jessie*, *Mosquitto*, *Python*, *MongoDB Database*. Pada

penelitian ini *Server Broker* menjadi penengah antara objek dan pengguna yang dapat menyimpan dan menyampaikan data yang diinginkan oleh user.

Alur kerja dari *Server Broker* terdiri dari beberapa tahapan yang berlangsung di dalamnya. Agar dapat bekerja seperti yang diinginkan pengaktifan *Broker Mosquitto* dan *PubSub_Middleware* harus dilakukan terlebih dahulu. Setelah aktif, sistem dapat bekerja dengan alur yang telah dirancang yaitu dapat menjadi jembatan antara sensor node dan *subscriber* untuk *live data*. Dapat menyimpan *live data* dengan sistem *sub_middleware* yang menjadi jembatan untuk *database MongoDB*. Dapat melakukan *publish* *live data* dengan sistem *pub_middleware* dari data *live* yang telah disimpan.

5.1.2.1 Alur kerja *Server Broker Real Time*

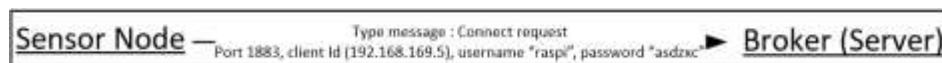
Pada perancangan ini menjelaskan tentang gambaran alur kerja *server broker* yang melakukan aktifitas penerimaan data dari sensor secara *real time*. *Real time* disini dimaksudkan penerimaan data *publisher* dari sensor yang langsung diteruskan kepada *subscriber*. Tujuan dari fitur ini adalah untuk memudahkan user memantau kondisi pH Air diluar waktu yang ditentukan. Berikut gambar 5.4 alur kerja *server broker real time live data*.



Gambar 5.4 Alur Kerja *Real Time Live Data*

Penjelasan dari Gambar 5.4 sebagai berikut:

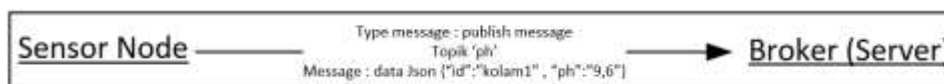
1. SensorNode meminta *connect & Authentikasi* kepada broker.



2. *Broker* memberi respon terhadap sensor node dan terjalin koneksi.



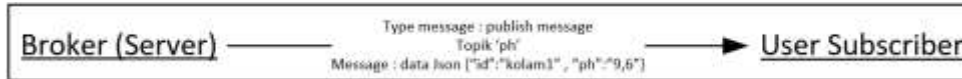
3. SensorNode melakukan *publish* *live data* kepada broker dengan *topik* dan *file Json* yang berisi *identitas* dan nilai pH.



4. *Broker* menerima koneksi dari *pengguna subscriber* dan memberi respon koneksi.

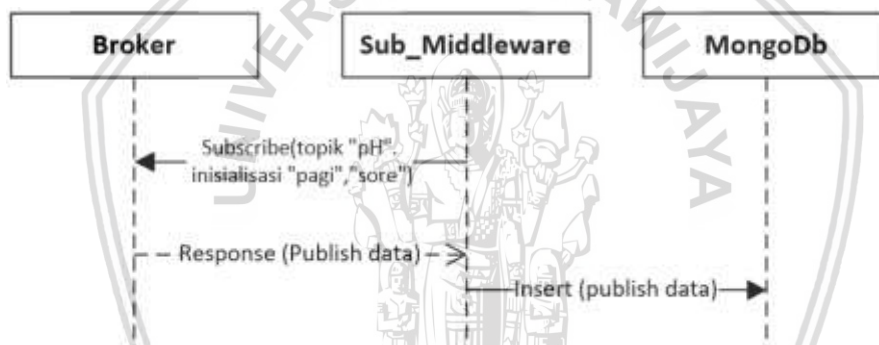


5. *Broker* memberi balasan dengan mengirimkan live data dari *publisher*.



5.1.2.2 Alur kerja Server Broker Subscribe Middleware to Database

Pada perancangan ini menjelaskan tentang gambaran alur kerja *server broker* yang melakukan aktifitas penyimpanan data dari sensor dengan fitur *sub_Middleware ke Database*. Fitur ini dilakukan oleh *sub_Middleware* dengan topik dan waktu yang telah ditentukan. Kemudian data yang telah didapatkan oleh *sub_Middleware* akan melakukan *insert* data pada *mongoDB* sebagai *database* yang digunakan. Tujuan dari fitur ini adalah untuk memudahkan user menyimpan data yang dari sensor untuk nantinya bisa di buat sebagai acuan jika terjadi sesuatu. Berikut gambar 5.5 alur kerja *sub_Middleware to Database*.



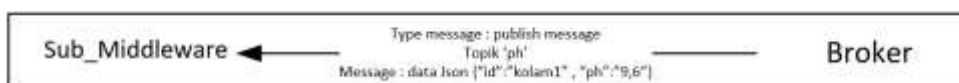
Gambar 5.5 Alur Kerja Subscriber Middleware to Database

Penjelasan dari Gambar 5.5 sebagai berikut:

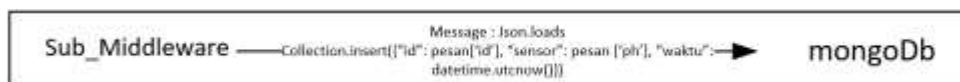
1. *Sub_Middleware* akan melakukan *subscriber* pada waktu yang ditentukan yaitu pukul 07.00-07.10 pagi dan pukul 16.00-16.10 sore.



2. *Broker* melakukan *respons* dengan memberi live data pada waktu yang telah ditentukan oleh *Sub_Middleware*.

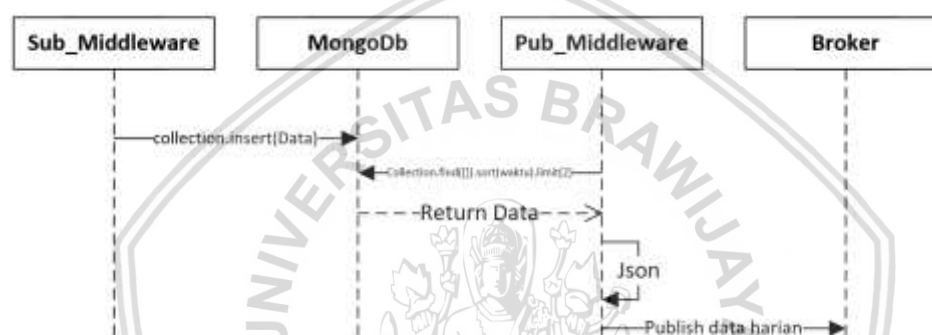


3. *Sub_Middleware* menerima live data dari *broker* dan meneruskan kepada *MongoDb Database*



5.1.2.3 Alur Kerja Server Broker Pub_Middleware Harian

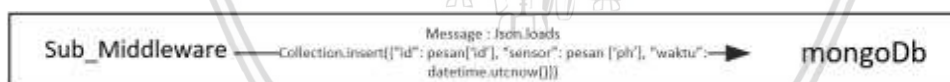
Pada perancangan ini menjelaskan tentang gambaran alur kerja *server broker* yang melakukan aktifitas pengiriman data dari database dengan topik “harian” berupa *publisher*. Pada fitur ini piranti *pub_Middleware* melakukan pengambilan data dari database dengan limit data yang bisa diambil. *Pub_Middeware* melakukan *collection.find({}).sort(waktu).limit(2)* yang artinya hanya mengambil data dari waktu dan batasan data yang bisa di ambi yaitu 2. Batasan tersebut bisa dirubah tergantung kebutuhan sensor yang akan menjadi *publisher*. Tujuan dari fitur ini adalah untuk memudahkan user melihat data yang diambil dengan waktu yang ditentukan. Karena fitur ini mengambil data dari database yang telah disubscribe oleh *sub_Middleware*. Berikut gambar 5.6 Alur kerja *Pub_Middleware harian*.



Gambar 5.6 Alur Kerja *Pub_Middleware Harian*

Penjelasan dari Gambar 5.6 sebagai berikut:

1. Setelah *sub_middleware* meneruskan live data yang diperoleh kepada *MongoDb* dan mengupdate live data yang tersimpan.



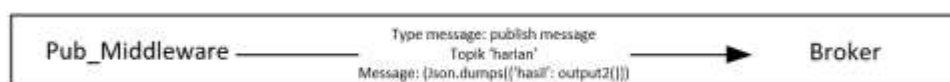
2. *Pub_Middleware* akan melakukan pengambilan data dengan *collect.find({}).sort(time).limit(2/4)* yang berarti mengambil 2 atau 4 data terakhir yang berada di *database*.



3. *MongoDB* melakukan *return data*.

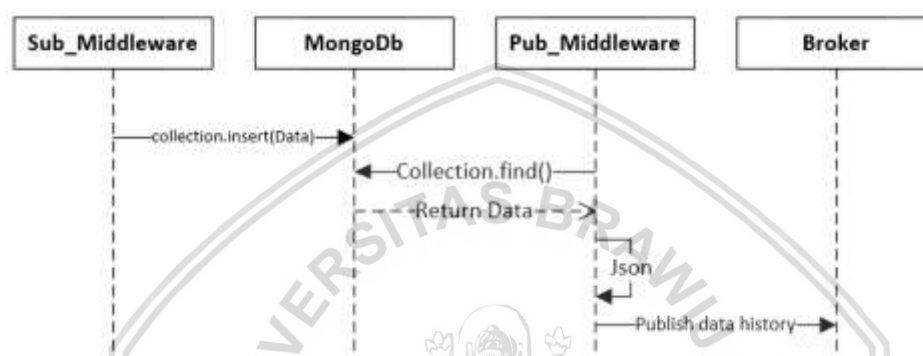


4. *Pub_Middleware_Harian* mempublish data tersebut yang dirangkai menjadi file *Json* kepada *Broker*.



5.1.2.4 Alur Kerja Server Broker Pub_Middleware History

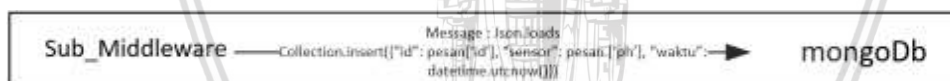
Pada perancangan ini menjelaskan tentang gambaran alur kerja *server broker* yang melakukan aktifitas pengiriman data dari *database* dengan topik "history" berupa *publisher*. Alur kerja yang diterapkan sama dengan *Pub_Middleware harian* tetapi memiliki perbedaan. Pada *Pub_Middleware history* pengambilan data dari *database* menggunakan *collection.find({})* dimana perintah tersebut untuk mengambil seluruh data tanpa limit yang berada di *database* dan kemudian dipublish. Tujuan dari fitur ini adalah untuk memudahkan user mengetahui masukan data dari sensor yang telah dipublish selama kurun waktu yang lama. Dengan demikian user dapat melakukan peninjauan dikemudian hari dengan hasil data yang telah tersimpan dari hari sebelumnya. Berikut gambar 5.7 Alur kerja *server broker Pub_Middleware history*



Gambar 5.7 Alur Kerja *Pub_Middleware History*

Penjelasan dari Gambar 5.7 sebagai berikut:

1. Setelah *sub_middleware* meneruskan live data yang diperoleh kepada *MongoDb* dan mengupdate live data yang tersimpan.



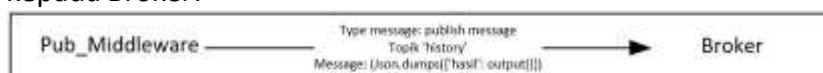
2. *Pub_Middleware* akan melakukan pengambilan data dengan *collect.find({})* yang berarti mengambil seluruh data yang berada di *database*.



3. *MongoDB* melakukan *return data*.



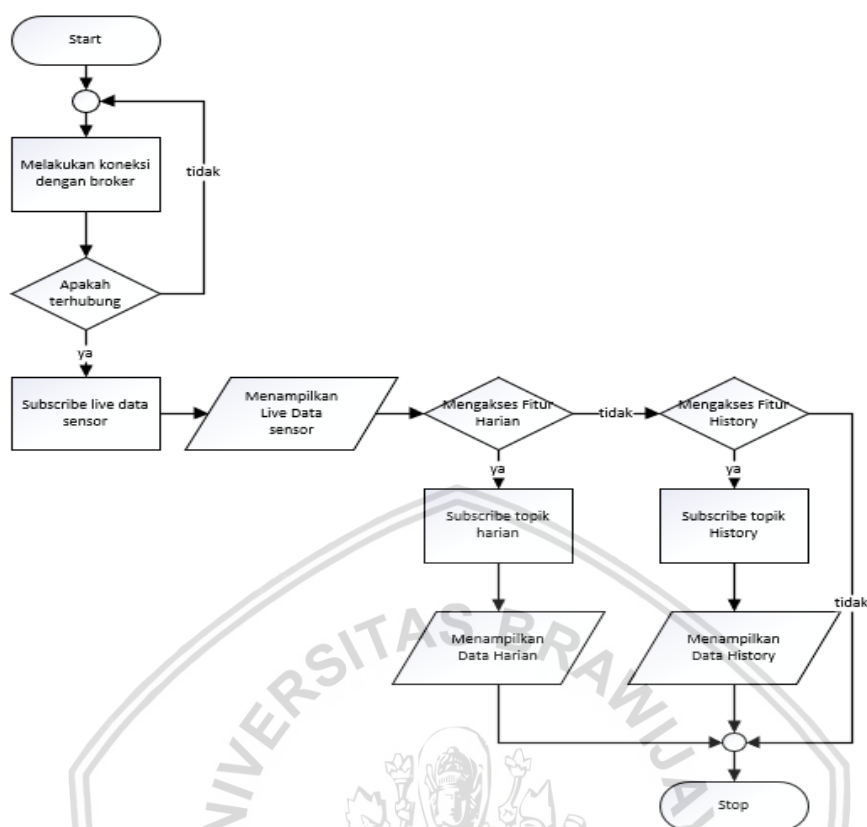
4. *Pub_Middleware_History* mempublish data tersebut yang dirangkai menjadi file *Json* kepada *Broker*.



5.1.3 Perancangan User Subscriber

Pada user subscriber ini user dapat melihat atau memantau hasil pengambilan data dari *sensor node*. User akan melihat live data yang berupa *realtime* dari *publisher*, melihat data yang telah dirancang untuk disimpan pada 2 waktu yaitu pagi dan sore, dan user dapat melihat hasil dari keseluruhan data dalam waktu yang telah berlalu atau *History*.

5.1.3.1 Alur Kerja User Subscriber



Gambar 5.8 Alur Kerja User Subscriber

Penjelasan dari Gambar 5.8 sebagai berikut:

1. *Pengguna Subscriber* melakukan koneksi kepada *Broker*
2. *Pengguna Subscriber* terhubung dengan *Broker* dan menunggu response live data dari sensor node.
3. User dapat memilih fitur yang tersedia yaitu Harian dan History.
4. User memilih Harian, sistem akan melakukan *subscribe* pada *broker* dengan *topik* yang sama dengan *Pub_Middleware_Harian*.
5. *Broker* menerima dan merespon *request* dari *subscriber_harian*.
6. *Pengguna Subscriber* menampilkan hasil dari response.
7. User memilih History, sistem akan melakukan *subscribe* pada *broker* dengan *topik* yang sama dengan *Pub_middleware_History*.
8. *Broker* menerima dan merespon *request* dari *subscribe_History*.
9. *Pengguna Subscriber* menampilkan hasil dari response.

5.1.4 Perancangan Database

Pada sistem ini menggunakan *database MongoDB* yang berbasis dokumen yang tentunya harus disesuaikan terhadap data yang akan disimpan dan *support data Json*. Dalam proses identifikasi memerlukan satu dokumen yang dimana dalam sistem ini bernama "*air*" untuk menyimpan informasi dari *sensor node* seperti yang dijelaskan pada Gambar 5.9.

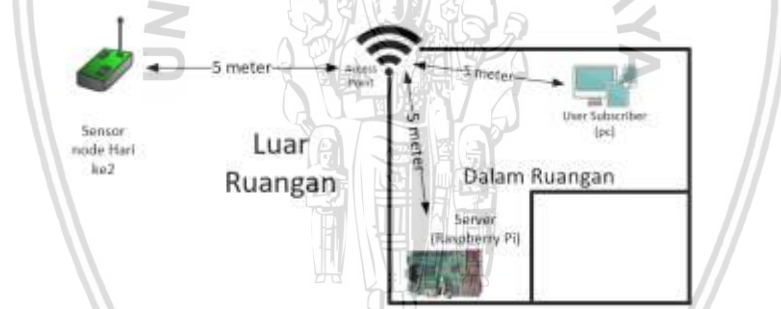
```
{
  "_id"      : ObjectId("5a55f89774fece0409004cc4"),
  "waktu"    : ISODate("2018-01-10T11:27:19.946Z"),
  "sensor"   : "9.11",
  "id"       : "kolam1"
}
```

Gambar 5.9 Skema Collection MongoDB

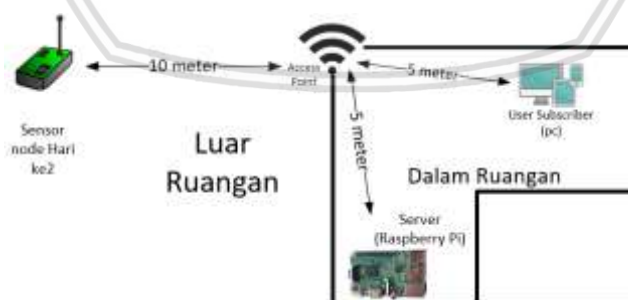
Gambar 5.9 merupakan rangkaian yang digunakan dalam sistem yang menggunakan konsep *key-Value* dimana setiap dokumen/*record* pada *database* memiliki *key*.

5.1.5 Lingkungan Pengujian

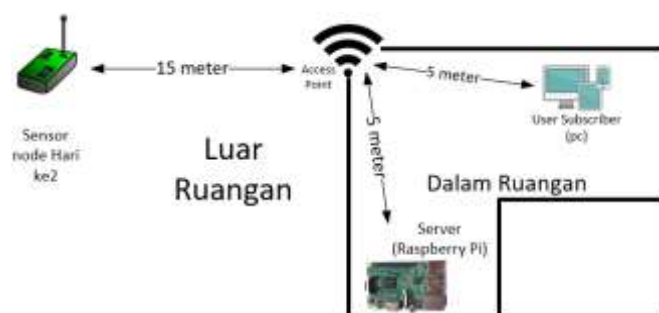
Tahap ini merupakan perancangan dengan suasana lingkungan. Dimana penempatan setiap *device* ditempatkan pada posisi yang telah diatur seperti pada rancangan pengujian cakupan area dengan lingkungan atau suasana yang sebenarnya. Dimana posisi sensor node berada pada ruang terbuka dengan cakupan area 5 meter dihari ke1, 10 meter dihari ke 2 dan 15 meter dihari ke 3. Tujuan dari penempatan sensor node diluar ruangan agar mengetahui nilai keaslian dari parameter uji yang telah dicantumkan. Kemudian untuk *access point* ditempatkan pada kisi-kisi ruangan agar dapat memancarkan sinyal untuk menjembatani *sensor node*, *server*, *subscriber*. Untuk penempatan *server* dan *subscriber* ditempatkan didalam ruangan dengan jarak masing-masing 5 meter dari *access point*. Berikut gambaran lingkungan pengujian yang dilakukan pada Gambar 5.10a.



Gambar 5.10a Pengujian Lingkungan 5 Meter Hari Ke 1



Gambar 5.10b Pengujian Lingkungan 10 Meter Hari Ke 2



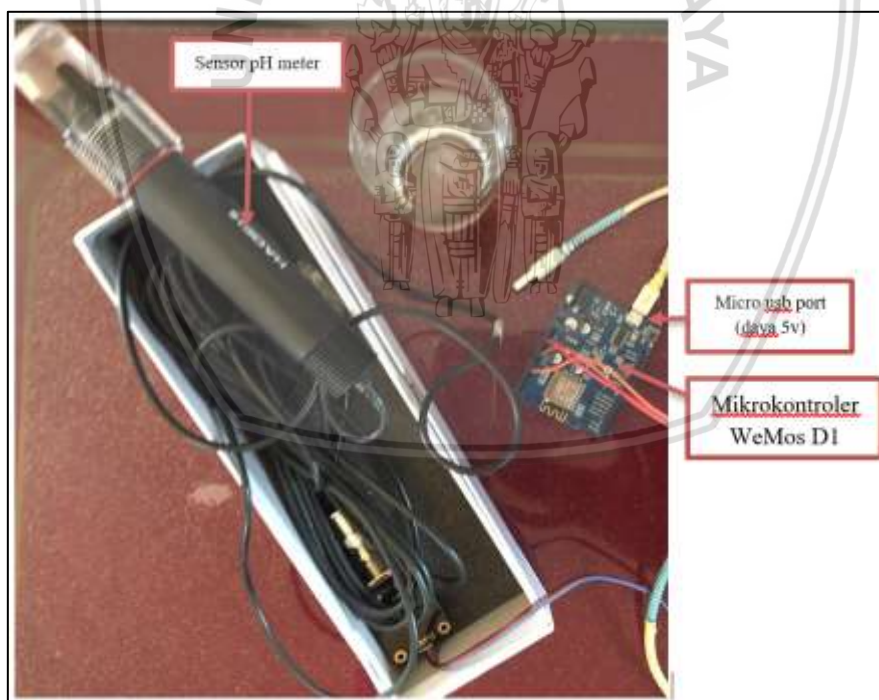
Gambar 5.10c Pengujian Lingkungan 15 Meter Hari Ke 3

5.2 Implementasi

Pada tahap ini implementasi sistem dilakukan ketika seluruh tahapan perancangan telah dipenuhi. Pada pembahasan implementasi sistem terdiri dari dua bagian, yaitu tentang implementasi perangkat keras dan implementasi perangkat lunak.

5.2.1 Implementasi Perangkat Keras Publisher

Implementasi perangkat keras, akan disesuaikan dengan perancangan yang telah dilakukan sebelumnya. Dengan membangun konsep sensor node yang nantinya akan berfungsi untuk mengoleksi data pH air dan dapat mengirimkan pada *server broker*.



Gambar 5.11 Implementasi Sensor Node Publisher

Pada gambar 5.11 dapat dilihat hasil implementasi perangkat keras pada sistem pengamatan pH air dengan sensor yang terhubung dengan mikrokontroler *Wemos D1* menggunakan kabel *jumper*. Untuk konfigurasi pin – pin pada masing-masing alat telah disesuaikan pada tabel keterangan pin yang telah dijelaskan pada tahapan perancangan. Untuk menyalakan sensor node terdapat sebuah *port micro usb* yang dapat digunakan

menggunakan daya listrik dengan kisaran besaran 3.5-5V yang dapat mengaplikasikan sebuah adaptor 5V maupun pengisi daya portable.

5.2.2 Implementasi Perangkat Lunak Publisher

Pada perangkat lunak implementasi sistem terbagi menjadi tiga bagian, diantaranya adalah bagian *publisher*, *broker*, dan *subscriber*.

5.2.2.1 Implementasi Sensor Node

Pada implementasi ini diperlukan sebuah *script* yang nantinya berfungsi untuk dijalankan pada sensor node. Pembuatan *script* ini membutuhkan *software* untuk mengaturnya yaitu menggunakan aplikasi *Arduino IDE* karena penelitian ini menggunakan mikrokontroler *Arduino Wemos D1*. *Script* yang dibuat hanya untuk menjalankan protokol MQTT sebagai *publish* dan *code* sensor *pH Probe Meter* sebagai pembaca objek. Pertama yang harus digunakan adalah library "*PubSubClient.h*", yang mana library tersebut menerapkan konsep komunikasi MQTT yang digunakan pada *Arduino*. Kedua menggunakan library "*ESP8266WiFi.h*" karena mikrokontroler yang digunakan telah didukung oleh modul *chip ESP8266* agar dapat digunakan dengan koneksi WiFi. Ketiga library "*ArduinoJson.h*" yang berfungsi sebagai pengubah data yang dikirim nanti menjadi data *Json*. Berikut pengimplementasiannya pada tabel 5.2.

Tabel 5.2 Implementasi Program Library

1	#include <ESP8266WiFi.h>
2	#include <PubSubClient.h>
3	#include <ArduinoJson.h>

A. Implementasi Program melakukan Koneksi pada MQTT Server

Terlebih dahulu *Publisher* menginisialisasi jaringan WiFi dengan menerapkan SSID dan Password yang digunakan. Kemudian menginisialisasi MQTT Server pada broker dengan IP Server yang digunakan agar *publisher* dapat terikat dengan *Broker*. Dan menginisialisasikan Topik yang akan digunakan supaya *Subscriber* mudah mendapat kan data yang diinginkan. Berikut implementasinya tabel 5.3.

Tabel 5.3 Implementasi Inisialisasi untuk menghubungkan dengan WiFi dan MQTT Server

1	const char* ssid = "KS_747R_42fd";
2	const char* password = "12345qwe";
3	const char* mqtt_server = "192.168.169.26";

B. Implementasi Program melakukan Publisher

Setelah tahap sebelumnya kemudian dilakukan *Publisher* mengacu pada topik yang telah digunakan untuk mendapatkan nilai dari sensor yang telah format dalam data *Json* dan kemudian akan dikirim ke *Broker* dan ditampilkan pada *Subscriber*. Berikut penggalan implementasiannya pada tabel 5.4.

Tabel 5.4 Implementasi inialisasi untuk publish dengan data Json

```

1 StaticJsonBuffer<300> JSONbuffer;
2 JsonObject& JSONencoder = JSONbuffer.createObject();
3 JSONencoder["id"] = "kolam1";
4 JSONencoder["ph"] = String(phValue);
5 char JSONmessageBuffer[100];
6 JSONencoder.printTo(JSONmessageBuffer,
7 sizeof(JSONmessageBuffer));
8 // char json =
9 "{\"id\":\"kolam1\",\"ph\":\""+String(phValue)+"\"}";
10 Serial.print(JSONmessageBuffer);
11 client.publish("ph", JSONmessageBuffer, true);
12 digitalWrite(13, HIGH);
13 delay(800);
14 digitalWrite(13, LOW);
15 Serial.println();

```

C. Implementasi Program *looping* koneksi 8menit

Tahap ini menjelaskan tentang *looping* koneksi setelah sensor node berhasil *publish* kepada *broker* dalam *server*. Setelah berhasil maka sensor node akan memutuskan koneksinya secara otomatis dan kemudian akan kembali memulai koneksi setelah 8 menit terputus. Berikut penggalan kode implementasi pada tabel 5.5.

Tabel 5.5 Implementasi on/off koneksi 8 menit

```

1 void offwifi(){
2   delay(5000);
3   WiFi.disconnect();
4   WiFi.mode(WIFI_OFF);
5   Serial.println("WIFI DC");
6   Serial.print(WiFi.status());
7   Serial.println();
8 }
9 void onwifi(){
10  delay(478000);
11  WiFi.mode(WIFI_STA);
12  WiFi.begin(ssid,password);
13  while (WiFi.status() != WL_CONNECTED) {
14    delay(500);
15    Serial.print("RECONNECTING");
16    Serial.println();
17  }
18  Serial.println("");
19  Serial.println("WiFi connected");
20  Serial.println("IP address: ");
21  Serial.println(WiFi.localIP());
22 }
23
24 void loop() {
25   while (WiFi.status() != WL_CONNECTED) {
26     delay(500);
27     onwifi();
28   }
29   if (!client.connected()) {
30     reconnect();
31   }
32   client.loop();

```



```

Es_747R_42fd
RECONNECTING
RECONNECTING
RECONNECTING
RECONNECTING
RECONNECTING
RECONNECTING
RECONNECTING
RECONNECTING
RECONNECTING

WiFi connected
IP address:
192.168.169.5
Attempting MQTT connection...connected
pH:10.60
{"id":"kolam1","ph":"10.60"}
WIFI DC
6

```

Gambar 5.12 Implementasi Hasil *node sensor*

Pada gambar 5.12 terlihat bahwa pertama sensor node melakukan koneksi kepada *access point* yang digunakan sebagai area komunikasi sistem. Setelah terhubung sensor node melakukan koneksi pada *server broker* dengan *inisialisasi autentikasi* yang telah disematkan. Kemudian setelah terhubung sensor node akan melakukan pengiriman data json dengan cara mempublish kepada *broker*. Jika telah dinyatakan *publish* berhasil sensor node akan memutuskan koneksi kepada *access point* maupun *server broker*. Sensor node akan melakukan *looping* setelah 8 menit dari kondisi terputusnya hubungan dengan *server*.

5.2.2.2 Implementasi Server Broker

Pada implementasi *Server Broker*, *raspberry pi* terlebih dahulu harus melalui proses *update* dan *upgrade* untuk mendapatkan versi terbaru. Kemudian menginstall *Python* untuk sarana pembangunan *PubSub_Middleware*. Berikut *code* untuk melakukannya pada tabel 5.6.

Tabel 5.6 Update, upgrade dan install python

1	Sudo apt-get update
2	sudo apt-get upgrade
3	sudo apt-get install Python
4	sudo apt-get install Python-pip

Setelah proses *update*, *upgrade* dan installasi *python* dilakukan kemudian membuat sebuah *script* yang berfungsi sebagai perantara *database* dan komunikasi terhadap user yaitu *PubSub Middleware*. Berikut *script PubSub Middleware* yang digunakan pada tabel 5.7.

Tabel 5.7 Script PubSub Middleware

1	from datetime import datetime, time
2	import time, json, pymongo
3	import paho.mqtt.client as mqtt
4	import paho.mqtt.publish as publish
5	
6	ip = "127.0.0.1"
7	topik = "ph"
8	
9	con = pymongo.MongoClient('127.0.0.1',27017)
10	collection = con.sensor.air


```

11
12 def on_connect(client, userdata, flags, rc):
13     print("KONEKSI SUKSES "+str(rc))
14     client.subscribe(topik)
15
16 def on_message(client, userdata, msg):
17     now = datetime.now()
18     now_time = now.time()
19     time_start = datetime.strptime('05/07/2018 00:00:30','%d/%m/%Y
20 %H:%M:%S')
21     time_end = datetime.strptime('05/07/2018 00:10:00','%d/%m/%Y
22 %H:%M:%S')
23     mor_start = time_start.time()
24     mor_end = time_end.time()
25     time_start_eve = datetime.strptime('29/06/2018
26 16:00:00','%d/%m/%Y %H:%M:%S')
27     time_end_eve = datetime.strptime('29/06/2018
28 16:10:00','%d/%m/%Y %H:%M:%S')
29     even_start = time_start_eve.time()
30     even_end = time_end_eve.time()
31     print now_time
32     if mor_start <= now_time <= mor_end:
33         print "PAGI"
34         print(msg.topic+" "+msg.payload)
35         if msg.topic=="ph":
36             pesan = json.loads(msg.payload)
37
38     collection.insert({"id":pesan['id'], "sensor":pesan['ph'], "waktu":
39     datetime.utcnow()})
40
41     elif even_start <= now_time <= even_end:
42         print "SORE"
43         print(msg.topic+" "+msg.payload)
44         if msg.topic=="ph":
45             pesan = json.loads(msg.payload)
46
47     collection.insert({"id":pesan['id'], "sensor":pesan['ph'], "waktu":
48     datetime.utcnow()})
49     else:
50         print "NOT IN TIME"
51
52 def on_publish(mosq, obj, mid):
53     print("Pub : " + str(mid))
54
55 client = mqtt.Client()
56 client.on_connect = on_connect
57 client.on_message = on_message
58 client.username_pw_set("raspi","asdzxc")
59 client.connect(ip, 1883, 60)
60 client.loop_start()
61
62 while True:
63     #publish history
64     output = []
65     for h in collection.find():
66         h['_id']= str(h['_id'])
67         h['waktu']= str(h['waktu'])
68         output.append(h)
69     client.publish("history", json.dumps({'hasil':output}))
70
71     #publish empat data terakhir dari db
72     output2 = []
73     for i in collection.find({}).sort([("waktu",-1)]).limit(2):
74         i['_id']= str(i['_id'])
75         i['waktu']= str(i['waktu'])
76         output2.append(i)
77     client.publish("harian", json.dumps({'hasil':output2}))
78     time.sleep(10)

```

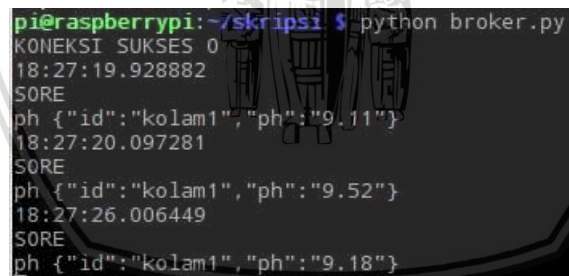
Penjelasan dari potongan kode diatas:

1. Baris ke 1-4 melakukan konfigurasi library yang dibutuhkan.
2. Baris ke 6-7 melakukan konfigurasi ip dan topik.
3. Baris 9-10 inisialissai kepada *MongoDB*.
4. Baris ke 12-50 melakukan konfigurasi untuk melakukan *subscriber* dengan instruksi pengambilan live data dengan waktu dan kemudian mengirimkannya kepada *MongoDB* untuk disimpan.
5. Baris ke 52-78 melakukan inialisasi konfigurasi *publisher* dengan pengambilan data di *database* dan dibagi antara 2 teratas dan keseluruhan data yang berada di *database*.



```
Berkas  Sunting  Tab Bantuan
pi@raspberrypi:~/skripsi $ python broker.py
KONEKSI SUKSES 0
18:25:47.108756
NOT IN TIME
18:25:51.182949
NOT IN TIME
18:25:57.128408
NOT IN TIME
```

Gambar 5.13 Pengaktifan *Broker.py* dengan skema tidak pada waktunya menyimpan



```
pi@raspberrypi:~/skripsi $ python broker.py
KONEKSI SUKSES 0
18:27:19.928882
SORE
ph {"id":"kolam1","ph":"9.11"}
18:27:20.097281
SORE
ph {"id":"kolam1","ph":"9.52"}
18:27:26.006449
SORE
ph {"id":"kolam1","ph":"9.18"}
```

Gambar 5.14 Pengaktifan *Broker.py* dengan skema pada waktunya menyimpan

Pada gambar 5.13 dapat dilihat bahwa *server* menampilkan aksi dari *sub_middleware* melakukan *subscribe* dengan topik yang digunakan sensor node. Pada saat menerima data *publish* dari *broker* namun tidak pada waktu yang ditentukan, maka akan tertera “not in time” yang berarti data yang didapat tidak akan *diinsert* pada *database*. Kemudian pada gambar 5.14 dapat dilihat, bahwa pada saat menerima data *publish* dari *broker* namun pada waktu yang ditentukan maka akan tertera waktu serta data yang akan *diinsert* pada *database*.

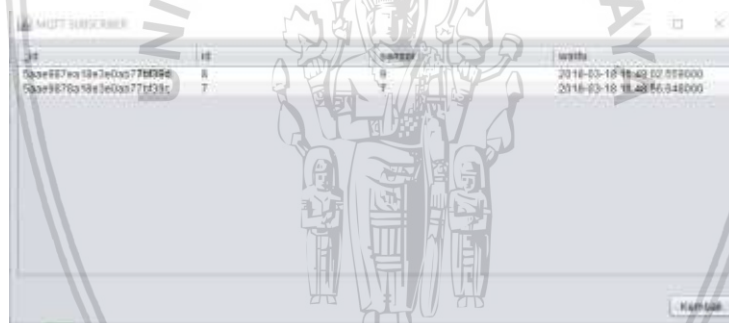
5.2.2.3 Implementasi User Subscriber

Subscriber menghubungkan pada *Broker* dengan IP MQTT Server juga Topik yang digunakan dan jika berhasil akan menerima data dari *Broker* yang sebelumnya diterima dari *Publisher*.



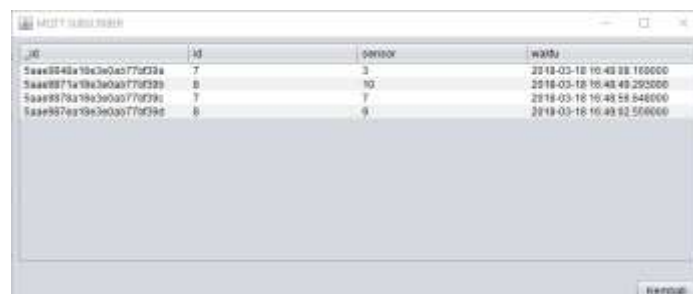
Gambar 5.15 Hasil implementasi *Subscriber Real Time*

Pada Gambar 5.15 merupakan *interface* untuk pengguna dalam pemantauan *real time* yang mana tujuannya agar user dapat mengetahui kondisi yang sedang terjadi. Dengan begitu user mengetahui kondisi diluar waktu pengecekan yang biasanya dilakukan.



Gambar 5.16 Hasil implementasi *Subscriber Harian*

Pada Gambar 5.16 merupakan *interface* untuk pengguna dalam pemantauan Harian yang mana bertujuan untuk melihat hasil dari pengamatan dalam waktu yang ditentukan.



Gambar 5.17 Hasil implementasi *Subscriber History*

Pada Gambar 5.17 merupakan *interface* pengguna untuk *History* yang bertujuan melihat keseluruhan yang telah disimpan pada database dalam dalam beberapa waktu. Dengan begitu user dapat melihat keseluruhan data yang telah didapatkan dalam waktu yang sebelumnya.



BAB 6 PENGUJIAN

6.1 Perancangan Pengujian

Perancangan pengujian dilakukan untuk membahas skenario dan parameter uji yang digunakan. Pada penelitian ini skenario pengujian dilakukan dengan 2 bagian yaitu pengujian fungsional dan pengujian protokol. Pada pengujian fungsional menggunakan parameter percobaan yang sesuai dengan perancangan sistem rancang bangun. Untuk pengujian kinerja pada protokol menggunakan parameter *latency*, *delay* dan *error rate*.

6.1.1 Perancangan Pengujian Fungsional

Pada perancangan pengujian fungsional dilakukan untuk mengetahui apakah rancang bangun sistem yang dibuat telah berjalan dengan benar. Dengan melihat tingkat keberhasilan pengujian melalui keluaran yang diperoleh dan mencocokkan dengan perancangan yang dibuat. Berikut tabel 6.1 skenario pengujian fungsional yang dilakukan.

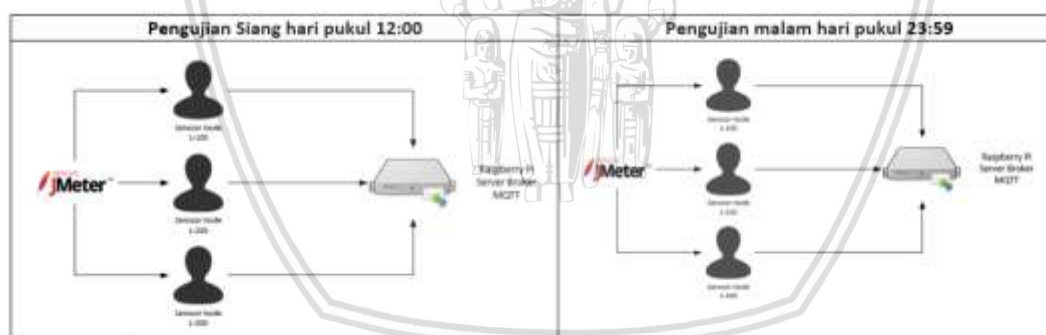
Tabel 6.1 Skenario Pengujian Fungsional

No	Deskripsi Pengujian	Skenario
PF_001	Pengujian sensor node <i>publisher</i>	<ol style="list-style-type: none"> 1. Melakukan koneksi dengan server broker. 2. Membaca sensor dan mengolah data. 3. Mempublish data dengan payload JSON.
PF_002	Pengujian server broker	<ol style="list-style-type: none"> 1. Menjalankan <i>middleware PubSub</i>. 2. Menjalankan database <i>MongoDb</i>.
PF_003	Pengujian <i>Middleware_Sub</i>	<ol style="list-style-type: none"> 1. Sensor node melakukan <i>publish</i> pada waktu yang ditentukan. 2. <i>Middleware_Subscriber</i> aktif, menerima data dan <i>insert</i> ke <i>MongoDB</i>.
PF_004	Pengujian <i>Middleware_Pub</i> harian	<ol style="list-style-type: none"> 1. Mengambil 4 data terakhir dari <i>mongodb</i>. 2. Mempublish dengan payload JSON.
PF_005	Pengujian <i>Middleware_Pub</i> history	<ol style="list-style-type: none"> 1. Mengambil seluruh data dari <i>mongodb</i>. 2. Mempublish dengan payload JSON.
PF_006	Pengujian Aplikasi User <i>Subscriber</i> live sensor node	<ol style="list-style-type: none"> 1. User mengsubscriber live sensor 2. <i>Middleware broker</i> merespon dan mengirimkan data live sensor

PF_007	Pengujian Aplikasi User <i>Subscriber</i> harian	<ol style="list-style-type: none"> 1. User mengirimkan <i>subscriber</i> dengan topik harian. 2. <i>Middleware broker</i> merespon dan mengirimkan data <i>middleware_pub</i> harian.
PF_008	Pengujian Aplikasi User <i>Subscriber</i> history	<ol style="list-style-type: none"> 1. User mengirimkan <i>subscriber</i> dengan topik history. 2. <i>Middleware broker</i> merespon dan mengirimkan data <i>middleware_pub</i> history.
PF_009	Pengujian kebenaran data sensor	<ol style="list-style-type: none"> 1. Menyamakan pH air dengan hasil yang telah didapatkan pada <i>subscriber</i>.

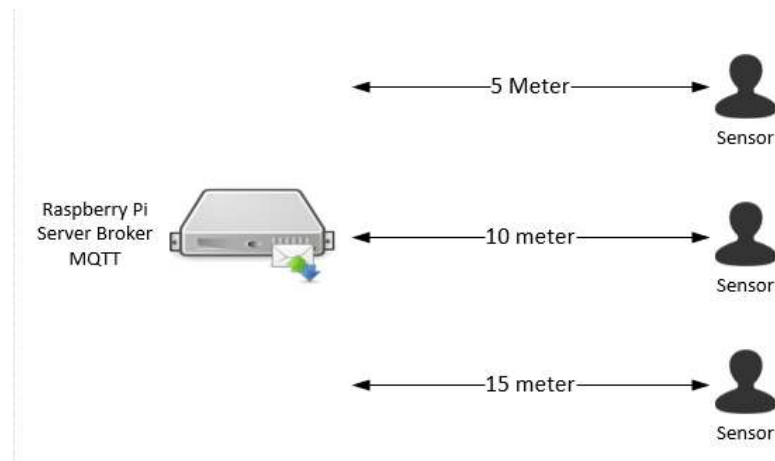
6.1.2 Perancangan Pengujian Kinerja Protokol

Pada perancangan pengujian kinerja protokol dilakukan dengan 2 sesi yaitu pada siang hari dan malam hari. Pengujian ini dilakukan untuk mengetahui besaran *latency* dan *error rate* yang didapat pada saat melakukan pengujian. Tahapan ini menggunakan bantuan *tool* atau perangkat untuk menjalankan skenario uji. Metode uji yang dilakukan menggunakan *tool testing Jmeter* sebagai perangkat *sensor node* dengan jumlah perangkat lebih dari 50 *publisher*.



Gambar 6.1 Rancangan Pengujian Kinerja protokol

Pada gambar 6.1 dapat dilihat pengujian kinerja pada protokol dibagi menjadi 2 sesi yaitu pengujian pada siang hari dan malam hari. Tujuan dari pembagian ini untuk mengetahui perbandingan besaran *latency* dan *error rate* yang diperoleh apakah terpengaruh oleh perubahan waktu yang berbeda. Pengujian dilakukan dengan skema membedakan banyaknya perangkat *sensor node* untuk melakukan *publish* bersamaan kepada *server*. Skema parameter yang digunakan adalah 100, 200 dan 300 perangkat *sensor node* melakukan *publish* kepada *broker* dalam satu waktu yang bersamaan. Jarak uji yang digunakan adalah 5 meter dari *Access Point (AP)*.



Gambar 6.2 Rancangan Pengujian Cakupan Area

Pada gambar 6.2 dapat dilihat pengujian dilakukan dengan jarak cakupan yang dapat digunakan. Pengujian ini untuk mengetahui besaran *Delay* yang diperoleh setiap jarak dalam komunikasi yang sedang berlangsung. Terdapat 3 jarak yang digunakan untuk melakukan uji cakupan area yaitu 5 meter, 10 meter dan 15 meter. Setiap jarak dilakukan pengujian selama 24 untuk mengetahui rata-rata *delay* yang diperoleh.

6.2 Hasil Pengujian dan Analisis

6.2.1 Hasil Pengujian dan Analisis Fungsional

Untuk mengetahui pengujian fungsional peneliti menggunakan 2 konsep penilai yaitu berhasil dan tidak berhasil. Dikatakan berhasil apabila fitur yang dirancang sukses dan sama seperti yang diharapkan tanpa error. Berikut hasil uji pengujian fungsional yang telah dilakukan pada tabel 6.2.

Tabel 6.2 Hasil Pengujian fungsional

No	Deskripsi Pengujian	Skenario	Hasil
PF_001	Pengujian sensor node <i>publisher</i>	4. Melakukan koneksi dengan server broker. 5. Membaca sensor dan mengolah data. 6. Mempublish data dengan payload JSON.	Berhasil
PF_002	Pengujian server broker	3. Menjalankan <i>middleware PubSub</i> . 4. Menjalankan database <i>MongoDb</i> .	Berhasil
PF_003	Pengujian <i>Middleware_Sub</i>	3. Sensor node melakukan <i>publish</i> pada waktu yang ditentukan.	Berhasil

		4. <i>Middleware_Subscriber</i> aktif, menerima data dan <i>insert</i> ke <i>MongoDB</i> .	
PF_004	Pengujian <i>Middleware_Pub</i> harian	3. Mengambil 4 data terakhir dari <i>mongodb</i> . 4. Mempublish dengan payload JSON.	Berhasil
PF_005	Pengujian <i>Middleware_Pub</i> history	3. Mengambil seluruh data dari <i>mongodb</i> . 4. Mempublish dengan payload JSON.	Berhasil
PF_006	Pengujian Aplikasi User <i>Subscriber</i> live sensor node	3. User mensubscribe live sensor 4. <i>Middleware broker</i> merespon dan mengirimkan data live sensor	Berhasil
PF_007	Pengujian Aplikasi User <i>Subscriber</i> harian	3. User mengirimkan <i>subscriber</i> dengan topik harian. 4. <i>Middleware broker</i> merespon dan mengirimkan data <i>middleware_pub</i> harian.	Berhasil
PF_008	Pengujian Aplikasi User <i>Subscriber</i> history	1. User mengirimkan <i>subscriber</i> dengan topik history. 2. <i>Middleware broker</i> merespon dan mengirimkan data <i>middleware_pub</i> history.	Berhasil
PF_009	Pengujian kebenaran data sensor	1. Menyamakan pH air dengan hasil yang telah didapatkan pada <i>subscriber</i> .	Berhasil

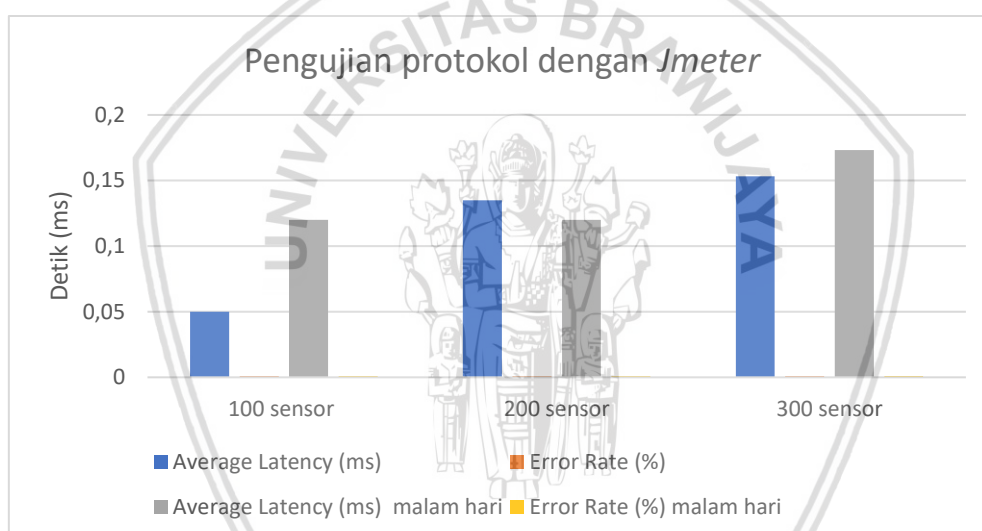
Pada tabel 6.2 dapat dilihat bahwa semua pengujian fungsionalitas sistem dinyatakan berhasil. Keberhasilan pada tahap ini dirujuk dari kesamaan pada tahap perancangan dan implementasi. Kemudian kesamaan tersebut dinilai sebagai apa yang diharapkan oleh peneliti.

6.2.2 Hasil Pengujian dan Analisis Kinerja protokol

Pada hasil pengujian kinerja dilihat dari 2 bagian yaitu hasil pengujian kinerja protokol dan hasil pengujian jarak cakupan area. Pertama hasil pengujian kinerja protokol yang dilihat besaran nilai yang didapat dari *latency* dan *error rate* yang diperoleh dari 2 waktu yang berbeda. Berikut hasil dari pengujian *protokol* dengan *Jmeter* pada tabel 6.3

Tabel 6.3 hasil Pengujian Kinerja protokol dengan *Jmeter*

Banyaknya sensor node	Siang Hari pukul 12:00		Malam Hari pukul 23:59	
	<i>Average Latency (ms)</i>	<i>Error Rate (%)</i>	<i>Average Latency (ms)</i>	<i>Error Rate (%)</i>
100	0,05	0	0,12	0
200	0,135	0	0,12	0
300	0,1533	0	0,1733	0



Gambar 6.3 Grafik pengujian kinerja protokol dengan *Jmeter*

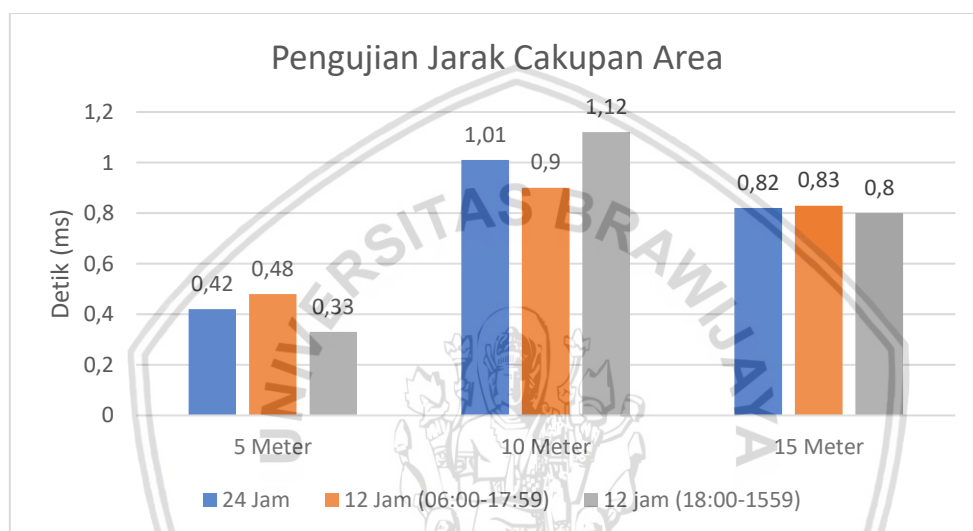
Dapat dilihat dari tabel 6.3 dan gambar 6.3 besaran nilai yang diperoleh dari kedua waktu yang berbeda memiliki perbedaan. Pada uji siang hari *Latency* yang diperoleh semakin banyak perangkat *sensor node* yang digunakan maka nilai yang didapat semakin tinggi. Berbeda dengan uji malam hari dimana *latency* yang diperoleh memiliki 2 nilai yang sama pada jumlah perangkat 100 dan 200 yaitu 0,12ms namun tidak pada perangkat 300 yang jauh lebih tinggi. Pada pengujian kinerja ini dapat ditarik kesimpulan bahwa perubahan waktu siang dan malam yang memiliki kondisi suhu dan angin yang berbeda sangat mempengaruhi waktu yang dibutuhkan untuk proses pertukaran data yang terjadi antara *client* dan *server* pada sistem.

Pengujian kedua yaitu cakupan area yang dapat digunakan untuk pengiriman data. Pengujian ini bertujuan untuk mengetahui cakupan area yang dilandasi dari lokasi sebenarnya. Peneliti menggunakan 3 jarak untuk mengetahui cakupan area untuk pengiriman

data yang mengacu pada parameter *delay*. Berikut hasil pengujian jarak yang telah dilakukan pada tabel 6.4.

Tabel 6.4 Hasil Pengujian Jarak Cakupan Area

Jarak	Average Delay (ms)		
	24 Jam	12 Jam (06:00-17:59)	12 Jam (18:00-05:59)
5 Meter	0,42	0,48	0,33
10 Meter	1,01	0,90	1,12
15 Meter	0,82	0,83	0,80



Gambar 6.4 Grafik Pengujian Jarak Cakupan Area

Pada tabel 6.4 dan gambar 6.4 dapat dilihat hasil dari pengujian sistem yang menitik fokuskan pada cakupan area dengan parameter delay. Hasil yang diperoleh pada pengujian ini memiliki nilai yang bervariasi. Dalam setiap jarak cakupan pengujian membedakan hasil nilai dengan 3 bagian yaitu selama 24 jam dan memfilter hasil tersebut menjadi 2 yaitu 12 jam siang pada pukul 06:00-17:59 dan 12 jam malam pada pukul 18:00-05:59.

Pada cakupan area 5 meter pada hari ke-1 dapat dilihat bahwa nilai *delay* yang diperoleh selama 24 jam adalah 0,42ms. Kemudian hasil tersebut selanjutnya difilter menjadi per-12 Jam sehingga diperoleh hasil 0,48ms pada 12 Jam (06:00-17:59) dan 0,33ms pada 12 Jam (18:00-05:59). Pada hari ke-1 kondisi cuaca pada siang hari cerah dengan suhu 34°C dan pada malam dengan suhu 24°C yang berangin.

Pada cakupan area 10 meter pada hari ke-2 dapat dilihat bahwa nilai *delay* yang diperoleh selama 24 jam adalah 1,01ms. Kemudian hasil tersebut selanjutnya difilter menjadi per-12 jam sehingga diperoleh hasil 0,90ms pada 12 Jam (06:00-17:59) dan 1,12ms pada 12 Jam (18:00-05:59). Pada hari ke-1 kondisi cuaca pada siang hari cerah mendung dengan suhu 34°C dan pada malam berawan dengan suhu 23°C yang berangin.

Pada cakupan area 15 meter pada hari ke-3 dapat dilihat bahwa nilai *delay* yang diperoleh selama 24 jam adalah 0,82ms. Kemudian hasil tersebut selanjutnya difilter menjadi per-12 Jam sehingga diperoleh hasil 0,83ms pada 12 Jam (06:00-17:59) dan 0,80ms pada 12 Jam (18:00-

05:59). Pada hari ke-1 kondisi cuaca pada siang hari cerah berawan dengan suhu 34°C dan pada malam berawan dengan suhu 24°C yang berangin. Dari ketiga hasil uji pada cakupan area untuk kebutuhan sistem yang dibuat dapat ditarik kesimpulan bahwa perubahan jarak dan perubahan cuaca berpengaruh pada proses pertukaran data yang terjadi antara *client* dan *server* pada sistem.



BAB 7 KESIMPULAN

Pada tahapan terakhir dari keseluruhan pembahasan yang telah dilakukan selanjutnya melakukan penarikan kesimpulan dan saran dari hasil penelitian yang telah dilakukan sebelumnya.

7.1 Kesimpulan

Dari hasil proses tahapan-tahapan yang telah dilakukan dalam melakukan perancangan, implementasi dan pengujian yang telah dilakukan, maka dapat ditarik kesimpulan sebagai berikut:

1. Dari hasil yang telah diperoleh, alur data dari node sensor menuji ke pengguna, yaitu ketika node sensor telah melakukan publish yang berisi data *json* dengan topik kepada *server*. Kemudian *server middleware* akan melakukan penyimpanan data ke database dan juga melakukan penginformasian data yang telah disimpan untuk *dipublish* kepada *broker*. Kemudian pengguna atau *subscriber* dapat melakukan *request* pada *server* untuk mendapatkan data *pH realtime*, *harian* dan *history*.
2. Mekanisme yang terjadi didalam *server* memiliki beberapa tahap yaitu, pertama *pub_sub middleware* yang telah aktif melakukan *subscribe* dengan topik yang sama pada *node sensor*. *Sub middleware* akan melakukan *insert_data* pada *database* pada saat waktu yang telah ditentukan, namun tidak akan melakukan *insert_data* jika tidak pada waktu yang ditentukan. Kedua untuk menginformasikan kepada pengguna terdapat *pub middleware* yang akan melakukan *get_data* pada *database*. Kemudian akan *dipublish* kepada *broker* dan oleh *broker* diteruskan kepada *subscriber* atau pengguna.
3. Dari hasil perancangan dan pengujian yang telah dilakukan, pengujian fungsional pada sistem telah berfungsi sesuai dengan yang diharapkan. Sedangkan pada pengujian kinerja protokol, dilihat dari nilai uji *latency*, *delay* dan *error rate*. Berikut penarikan kesimpulan untuk menguji kinerja protokol:
 - a. Dari hasil pengujian *latency* dan *error rate* menggunakan *Jmeter* dengan skenario uji 100, 200 dan 300 *client publisher* dalam waktu bersamaan kepada *broker* dan *pub_sub middleware* menghasilkan rata-rata *latency* dibawah *1ms* tepatnya *0,31ms* dengan *error rate* 0%.
 - b. Dari hasil pengujian *delay* untuk permasalahan cakupan area dengan skenario uji 5, 10 dan 15 meter jarak yang digunakan dalam kurun waktu 3x24 jam didapatkan nilai rata-rata dibawah *1ms* tepatnya *0,75ms*.
 - c. Dengan demikian penambahan fitur *pub_sub middleware* juga memiliki fungsi khusus yang berperan sebagai pengontrol dan perantara antara pengguna dan perangkat *IoT*. Tetapi penambahan fitur ini tidak membuat kinerja perangkat *IoT* maupun protokol mendapatkan mengalami permasalahan yang signifikan.

7.2 Saran

Berdasarkan dari hasil kesimpulan yang telah diperoleh, maka terdapat beberapa saran yang diharapkan berguna dalam mengembangkan sistem ini selanjutnya.

1. Pengembangan sistem *middleware* dengan protokol yang berbeda yang dapat digunakan pada teknologi *Internet of Things*. Dengan komunikasi yang mengutamakan antara *client* dan *server* untuk saling bertukar data.
2. Fitur *middleware* yang dapat diatur dari sisi *client*. Untuk memudahkan user mengubah aturan dalam *server*.
3. Fitur *security* yang dapat dikembangkan dengan lebih kompleks lagi.

Pengujian yang dapat dilakukan dengan aspek lainnya seperti melihat kinerja dari *server* dengan sistem *middleware* yang lebih kompleks dan protokol yang digunakan.



DAFTAR PUSTAKA

- Adi, H. K., P, E. S. & Amron, K., 2016. *PENGUMPULAN DATA MENGGUNAKAN METODE PUBLISH SUBSCRIBE PADA NODE SENSOR DALAM WIRELESS MESH NETWORK*. Volume 8.
- Ajeng, 2014. *Perubahan pH dalam Air sangat berpengaruh besar pada biota laut*. [Online] Tersedia di: <http://ukurkadarair.com/blog/perubahan-ph-dalam-air-sangat-berpengaruh-besar-pada-biota-laut> [Diakses pada 17 3 2017].
- Alsaadi, E. & Tubaishat, A., 2015. *Internet of Things: Features, Challenges, and Vulnerabilities. International Journal of Advanced Computer Science and Information Technology (IJACSIT)*., Volume 04.
- Asdary, M., 2017. *Permasalahan pemantauan pH air dan pemanfaatan teknologi IoT*. [Cahyo karya]. Balai Perikanan Budidaya Air Payau (BPBAP Situbondo). 27 Juli 2017, 10:23.
- Atzori, L., Iera, A. & Morabito, G., 2015. *The Internet of Things: A survey. e-journal*, 54 (2010)(15), pp. 2787-2805.
- Aziz, B., 2016 . *A formal model and analysis of an IoT protocol*. 36(P1), pp. 49-57.
- Barata, D., Louzada, G., Carreiro, A. & Damasceno, A., 2013. *System of acquisition, transmission, storage and visualization of Pulse Oximeter and ECG data using Android and MQTT*.
- dewi, p. m., 2017. *Mengenal Wemos D1 Mini Dalam Dunia IoT*. [Online] Tersedia di: <http://ilmuti.org/2017/02/23/mengenal-wemos-d1-mini-dalam-dunia-iot/> [Diakses pada 26 July 2017].
- Manyika, J. et al., 2015. *THE INTERNET OF THINGS: MAPPING THE VALUE BEYOND THE HYPE*, s.l.: McKinsey&Company.
- mohammadi, m. et al., 2015. *Internet of Things: A Survey on Enabling Technologies, Protocols and Applications*.
- Mufid, A., 2016. *INDUSTRI AKUAKULTUR UNTUK PERTUMBUHAN EKONOMI BERKELANJUTAN*. [Online] Tersedia di: <http://aquaculture-mai.org/archives/1643> [Diakses pada 26 12 2016].
- Nasa, D. S., 2015. *Panduan Pengelolaan Air Budidaya Ikan*. [Online] Tersedia di: <http://www.viternaplus.com/2015/09/panduan-pengelolaan-air-budidaya-ikan.html> [Diakses pada 17 03 2017].
- Pr, E., 2015. *Mengenal MQTT protokl IoT*. [Online] Tersedia di: <https://medium.com/pemrograman/mengenal-mqtt-998b6271f585> [Diakses pada 27 7 2017].
- Weiss, B. et al., 2016. *A Publish/Subscribe Messaging System For Wireless Sensor Communication*.

